The Chinese University of Hong Kong
Department of Computer Science and Engineering

LYU1802 - BotanWiki

Final Year Project
Final Report

Wong Tsz Hin 1155079510
Yan Chi Shing 1155078689

Supervised by: Prof. Lyu Rung Tsong Michael

Table of Contents

0. Abstract

Image recognition has an astonishing result in computer with the help of artificial Intelligence. The project has been kicked off and built an offline mobile application, a sustainable platform in mobile device to recognize plants by leaves in the Chinese University of Hong Kong, with a low error rate of recognizing the plants among the data we collected.

1. Introduction

1.1. Motivation

Being students studying computer science in CUHK, where is the most beautiful and scenes campus in Hong Kong, we always see different kind of trees and flowers shuttling in the school. Some of them have the peculiar appearance, some of them bloom the colorful flowers. Walking through the vivid campus. We might want to find out what kind of plant it is. Although some plants have already been labelled with the QR code, most of them do not have their own tag or lost the tag due to insufficient maintenance of the organization in charge.

AI technology nowadays have a wide range of research and usage, contributing to the society with influential impact. In the area of image recognition, neural network has the astonishing result of successfully recognizing object by training, for example a residual network has achieved recognizing the ImageNet test set, which has more than two hundred categories of object, only 3.57% error[1]. Therefore, we would like to make an attempt to build a mobile application, equipped with artificial Intelligence to recognition the plants in CUHK in a mobile way.

[1] Result is presented in the article discussing a residual learning approach for recognizing images, see the article "Deep residual learning for image recognition"

1.2. Significance of the project

Plants has a vital role in the life of human as well as the ecosystem. Distinguishing the species of plants over a massive category is challenging and expensive as it requires a comprehensive study of the plants, including the growing habits, the habitat, the particular characteristic of the stem, leaves and so on.

This project would like to provide a quick and inexpensive way to recognize plants by the leaf. It may be possible to reduce the workload of botanists when they are working on research with a massive data.

People in urban area can access internet with their mobile device easily, whereas people in rural area may not access the internet with the robust connection. In this circumstance, the mobile application would like to deliver and offline solution for people without internet connection to recognize plants in countryside.

Although some projects have already been initiated to recognize trees and plants in CUHK. This project would try to utilize neural network to contribute the recognition of plants with an alternative method.

1.3. Goal

The project is aimed at building a mobile application with artificial Intelligence in image recognition to distinguish plants without relying on internet access.

2. Technology Overview

2.1. TensorFlow

TensorFlow is an open source machine learning library for developers to implement a wide variety of machine learning techniques, from basic regression to classification. Google has also developed TensorFlow Lite, targeting at support machine learning in the mobile application development. Implementation of model and model visualization is used in this project.

2.2. OpenCV and Keras

OpenCV is the open source computer vision library for handling pictorial data in computer. In this project, we use to get do some file transformation and data augmentation for the training data. The version we used is OpenCV3.0

```python
import cv2
from glob import glob
from keras.preprocessing.image import ImageDataGenerator, array_to_img, img_to_array, load_img
import os

datagen = ImageDataGenerator(
        zca_whitening=False,
        rotation_range=40,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.5,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest')

pngs = glob('./**/*.png', recursive=True)
for j in pngs:


    fileName = os.path.basename(j)
    fileDirName = os.path.dirname(j)
    print("Now at" + fileDirName)
    print("Now in " + fileName)
    print("---")

    img = cv2.imread(j)  # this is a PIL image
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    #plt.imshow(img)
    #print(img.shape)
    img = img.reshape((1,) + img.shape)
    #print(img.shape)
    #x = img_to_array(img)  # this is a Numpy array with shape (3, 150, 150)
    #x = x.reshape((1,) + x.shape)  # this is a Numpy array with shape (1, 3, 150, 150)


    i = 0
    for batch in datagen.flow(img, batch_size=10, save_to_dir=fileDirName, save_prefix=fileName[:2],save_format='jpg'):
        i += 1
        if i > 4:
            break  # otherwise the generator would loop indefinitely
```

Figure 1: Data Augmentation of shearing using OpenCV and Keras

2.3. Android Studio

We would like to initiate the mobile application on android platform, since the many people are now using android phone. According to the statistic from IDC on smartphone market share, Android phone has 84.8% share, and iPhone have only 15.1% share[2].

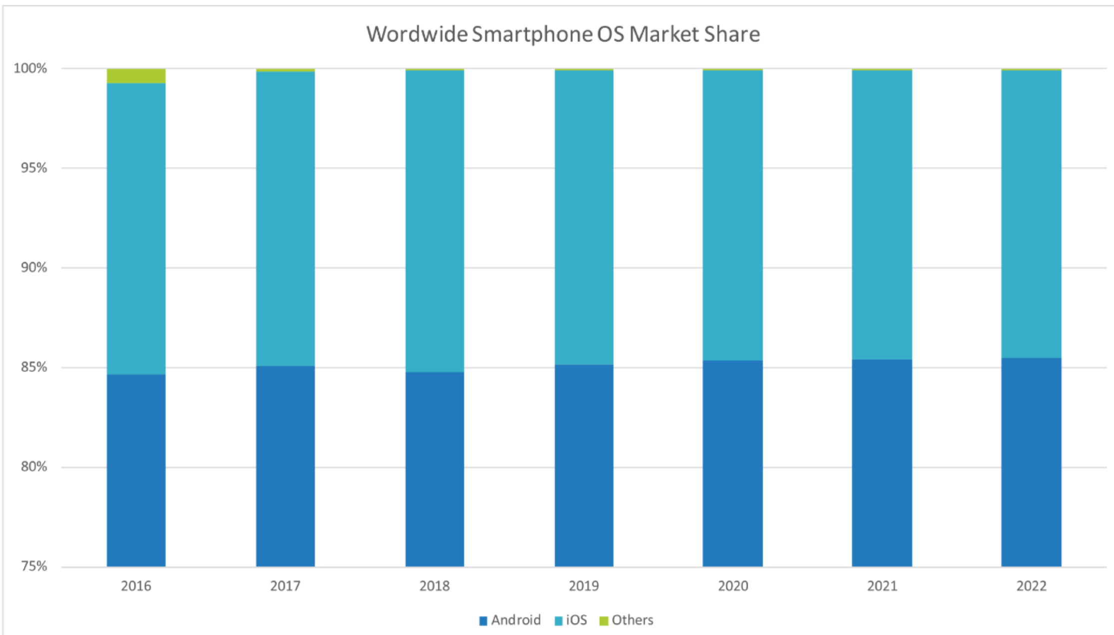


Figure 2: Statistic on Smartphone OS Market share from IDC

[2] Statistical Data obtain from IDC researching on smartphone market share.

Moreover, getting support of android-related issue from online is easier than getting support of ios-related issue, from the question counting in stackoverflow. Therefore, we choose our project in the android platform.

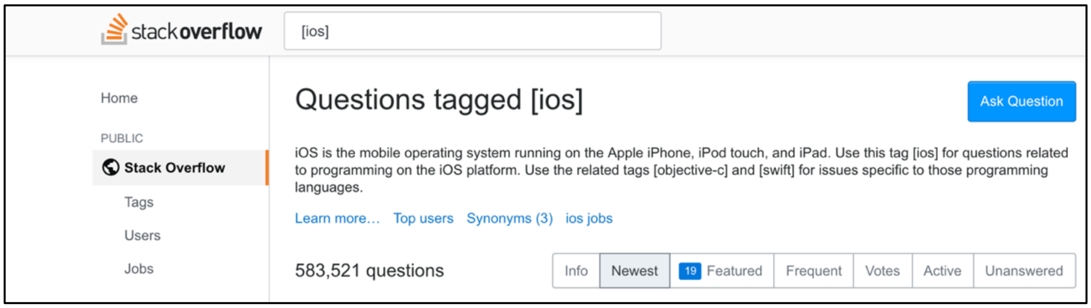


Figure 3: 583,521 questions have been asked about ios in stackoverflow

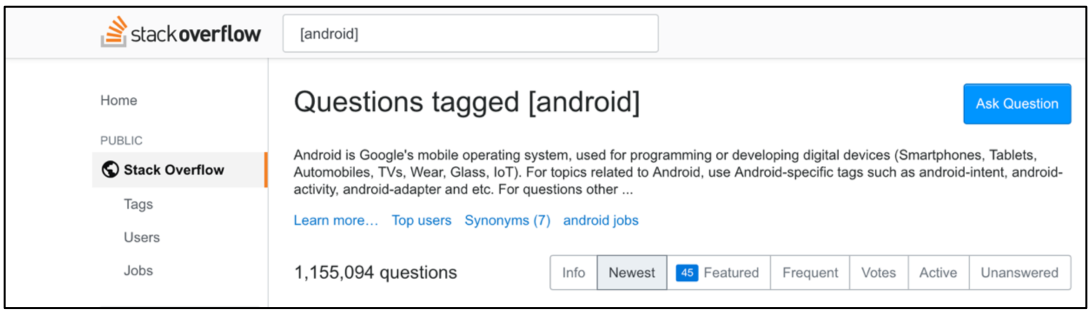


Figure 4: 1,155,094 questions have been asked about android in stackoverflow

Android Studio is the Android official Integrated Development Environment developed by Google. People who are interested in developing android application can take advantages from the IDE and build the app in a fast way.

2.4. AWS online server

Amazon Web Service is a big cloud computing platform, which provides a vast number of cloud service. In this project, the EC2 service would be used to obtain a virtual machine for server-sided implementation.

3. Preparation

3.1. Image Data

We have done some researchs about some database regarding leaves of trees, but we could not find a suitable data set for the project. As a result, we decided to collect the data by taking photos of the leaves by ourselves. With the guidance of the Curator of CUHK The Shiu-Ying Hu Herbarium, David Lau, we followed the online education website CUHK campus 100 plants[3], and take the picture of 12 species of tress/plants. However, in the experiments we conducted and the development of the application, we only chose 10 species to train our model. Sine "Araucaria heterophylla" and "podocarpus macrophyllus" are two plants that the leaves completely different from other plants' leaves (see figure below for comparison), we would to keep our experiment in a more controllable way. Therefore, we kept the data collected away and trained the model with the rest of 10.

scientific name	abbreviations	Herbarium URL / Wiki Url	Google Map	LatLng
Camellia granthamiana	cg	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet-pro/camelia-granthamiana/	https://goo.gl/maps/KbpuhDqAzRB2	22.415972,114.209083
canarium album	ca	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet-pro/canarium-album/	https://goo.gl/maps/8thiiw1dcJ22	22.415778,114.208917
handroanthus chrysanthus	hc	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet-pro/handroanthus-chrysanthus/	https://goo.gl/maps/VdqjFyvQQsj	22.416278,114.209278
Hibiscus rosa-sinensis	hr	https://en.wikipedia.org/wiki/Hibiscus_rosa-sinensis	https://goo.gl/maps/SauDbso9XLH2	22.415806,114.209333
camellia japonica	cj	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet-pro/camellia-japonica/	https://goo.gl/maps/vnPmbL6wsPB2	22.416028,114.209139
cinnamomum burmannii	cb	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet-pro/cinnamomum-burmannii/	https://goo.gl/maps/wfdnYDtzyuD2	22.415764,114.207528
bauhinia variegata	bv	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet-pro/bauhinia-variegata/	https://goo.gl/maps/w62A16aVv8L2	22.416923,114.206799
sterculia lanceolata	sl	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet/sterculia-lanceolata/	https://goo.gl/maps/9eTToGZHFwv	22.417028,114.206833
machilus chekiangensis	mc	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet-pro/machilus-chekiangensis/	https://goo.gl/maps/3KKggxvZJMn	22.417583,114.206861
ficus altissima	fa	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet/ficus-altissima/	https://goo.gl/maps/rSZJQqcr4S62	22.417830,114.205124
podocarpus macrophyllus	pm	http://syhuherbarium.sls.cuhk.edu.hk/collections/factsheet/podocarpus-macrophyllus/	https://goo.gl/maps/zetowL8DCdQ2	22.418000,114.205083
Araucaria heterophylla	ah	http://www2.cuhk.edu.hk/cuhktree/TreeDetails.aspx?tid=9&loc=Franklin	https://goo.gl/maps/cReQuGxR9G52	22.418083,114.204806

Figure 5: Excel Table show the plants we collected

[3] The website will show the map with plant marked in English name, http://syhuherbarium.sls.cuhk.edu.hk/collections/courseware/campus-100-plants/

Figure 6 : Araucaria heterophylla

Figure 7 : podocarpus macrophyllus

Figure 8 : Photo of "machilus chekiangensis"

Figure 9: Photo of " ficus altissima"

Figure 10 : Photo of "cinnamomum burmannii"

4. Specification of Dr.Leaf V1.0

4.1. Research on current market

Currently, there are some applications which are developed with the same purpose of performing the recognition on different plants with their photos. We have downloaded and tried out some of them, having the first-hand experience on the service provided by the mobile application. Also, we will evaluate, learn and take advantages from the application in the market.

Table 1: Summary of mobile applications

Name of Application	Availability of Platform	Origin	Features	Installation Count
PlantSnap	• IOS • Android	United State	• 585,000 species of plants which are available for classification in the database, with 90% [4] of accuracy • Supported Identification include plant, mushroom and cactuses • User Login system for further interaction between users on the plant identification	5,000,000+ in Google Play
PictureThis	• IOS • Android	China	• 4,000 species of plants which are available for classification in the	500,000+ in Google Play

[4] Statistical data such as the size of database and accuracy of identification mentioned in official website, https://www.plantsnap.com/

(形色 - 拍照識花識別植物)			database, with 98% [5] of accuracy • Response in 1 second for the recognition result • Online community for sharing recognition of plant	
Pl@ntNet	• IOS • Android	N/A	• 7657 species of plants in USA, 6700 species of plants in Western Europe[6] • No accuracy provided • User can indicate which part of the plant are taken in the photo, to increase the performance of recognition	5,000,000+ in Google Play

[5] Statistical data such as the size of database and accuracy of identification mentioned in official website, http://www.xingseapp.com/

[6] Statistical data such as the size of database and accuracy of identification mentioned in official website, https://identify.plantnet-project.org/

4.1.1. Highlight 1 – Online Database

We discover that all the mobile application we tested rely on the online database to do the recognition, so that they are able to perform the recognition over a massive species of plants. Although the source is large enough to cover as many as kind of plants, slow network will cause the bad performance on the claimed “instant recognition” and deliver unpleasant user experience to people who want to know the plant instantly.

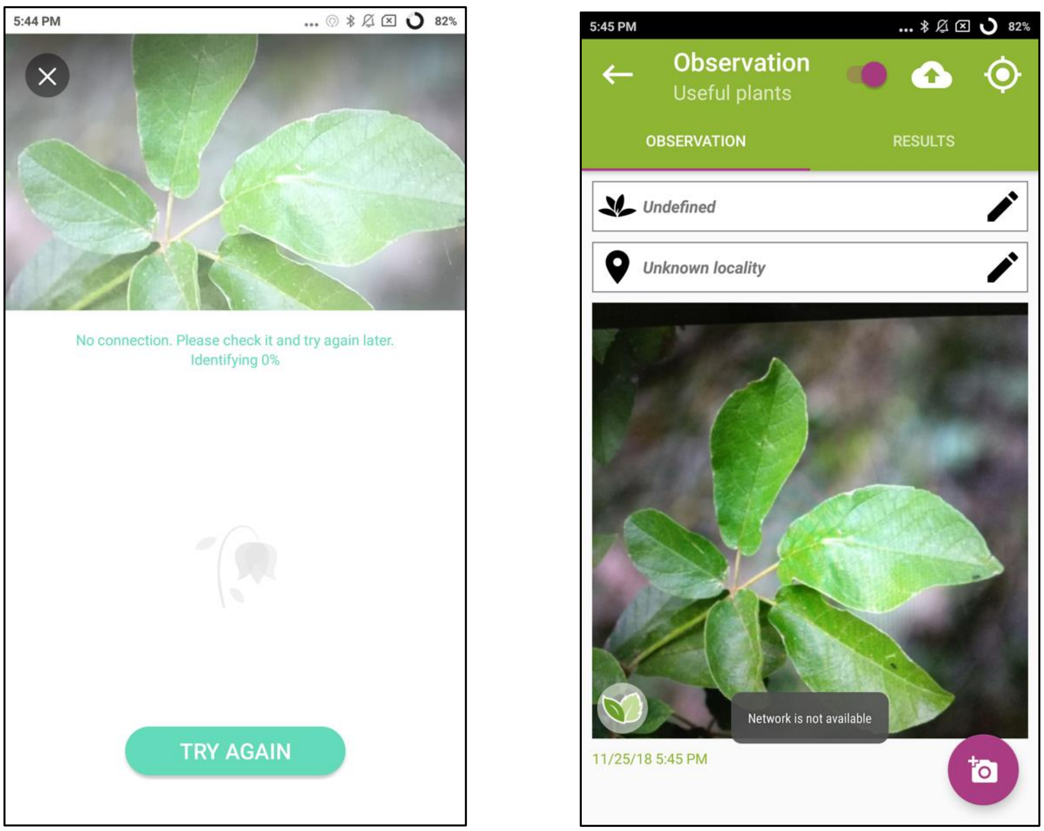


Figure 11 : Network Error in PictureThis and Pl@ntNet

As a result, we consider that user may always visit the countryside or some places where the network condition is unsatisfied, we will try to build the mobile application with offline database and do the recognition locally in the device.

4.1.2. Highlight 2 – Assistance of Recognition

In order to improve the recognition result, some of the application provide the extra feature tag or photo taking guidance for user. In pl@ntNet, it allows user to take picture of several parts of a plant, for example leaf, flower or fruit, and user have to indicate the part they are taking manually before doing the recognition. In PictureThis, user can shoot the picture with the provided focusing area, therefore the input picture to the modal will be standardized, having the similar format to the training data consisting of only leaves or being inside the valid area.

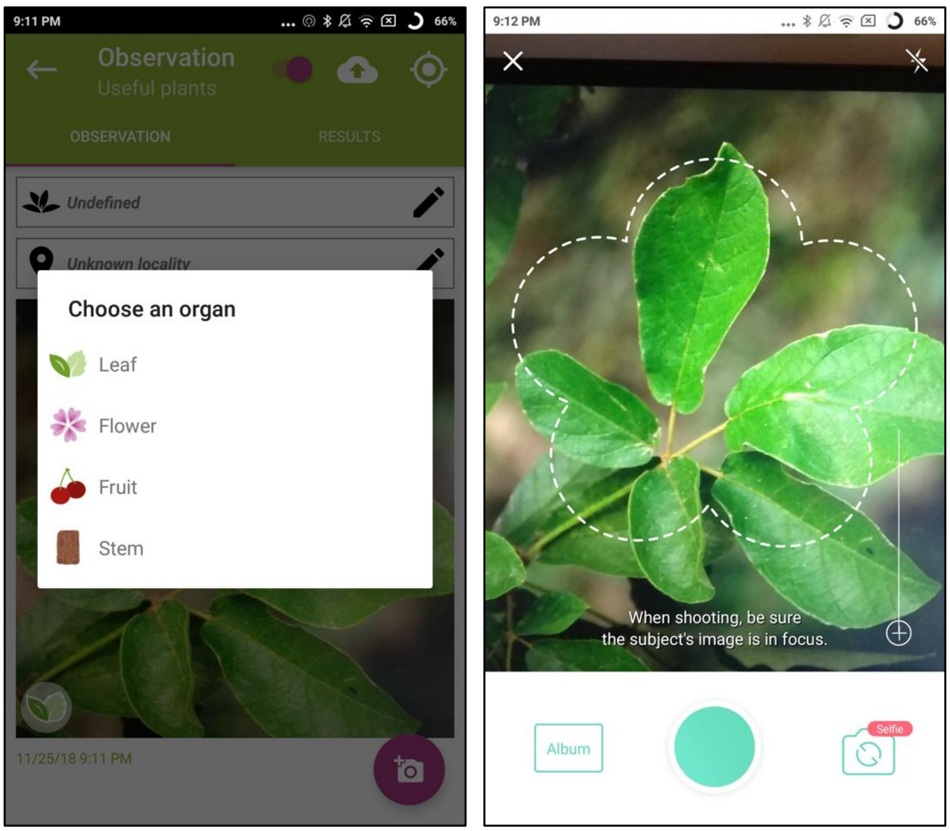


Figure 12 : Method using in pl@ntNet and PictureThis as assistance of recognition

It is a good way to control the input data with the guideline to the user, resulting in a more precise plant recognition. However, it may not be a user-friendly experience in the user perspective as they have to choose the option manually, in which will slow down the speed of getting the result from the application.

4.1.3. Highlight 3 – Community for user to share the plant photos

A large amount of training data is advantageous to an AI modal to get a satisfied recognition result. The application we found also set up a platform for user to share the photo they took with the label. Not only can users enjoy sharing the photo with others through the platform, they are able to rate others photo if they think it have a correct label. Developers can record the user behavior of rating others' photos and enhance the database of plant with the help of massive users.

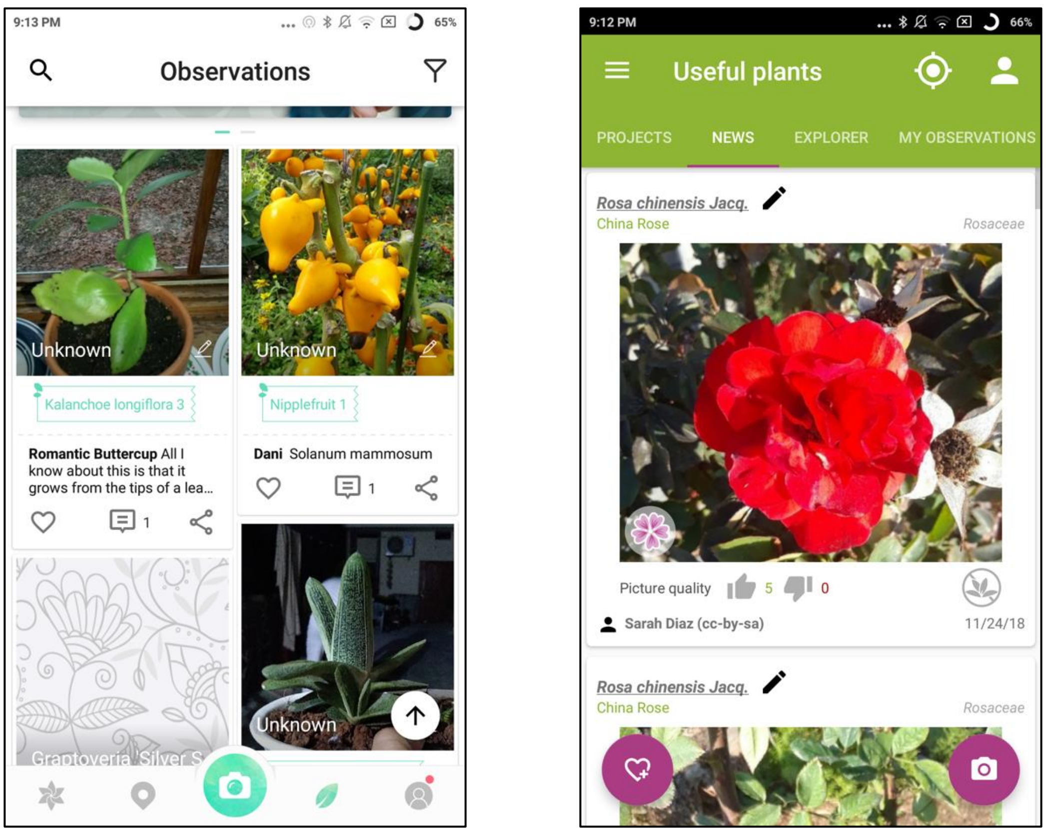


Figure 13 : Community for users in PictureThis and Pl@ntNet

This kind of community can result in win-win situation for both user and the developer, because the users can communicate and enjoy the interaction to others plant lovers who are using the same application. Handling massive leaf pictorial data is an arduous task for developer, this community can bring a great effort contributing to the mission of processing training data. The only concern of the community is that if the users do not have particular knowledge of the plant, they may make the wrong rating, causing the inappropriate training data for the model. Develop still cannot merely rely on the community to get the massive training data.

4.1.4. Highlight 4 – Development Issue

For the application "PlantSnap", we made an attempt to try and test the function of it. However, we are not able to take a single picture with the application. It comes the mobile application development issue regarding testing in different devices. Portability of the application is the main concern of developer, since there are at least five important platform with enormous model of mobile phone respectively. Developers are not possible to test the application among all the device with several operation systems, and we are suggested to keep the basic and common feature and components that are available across in all platforms and build up the application upon those variants and components[7].

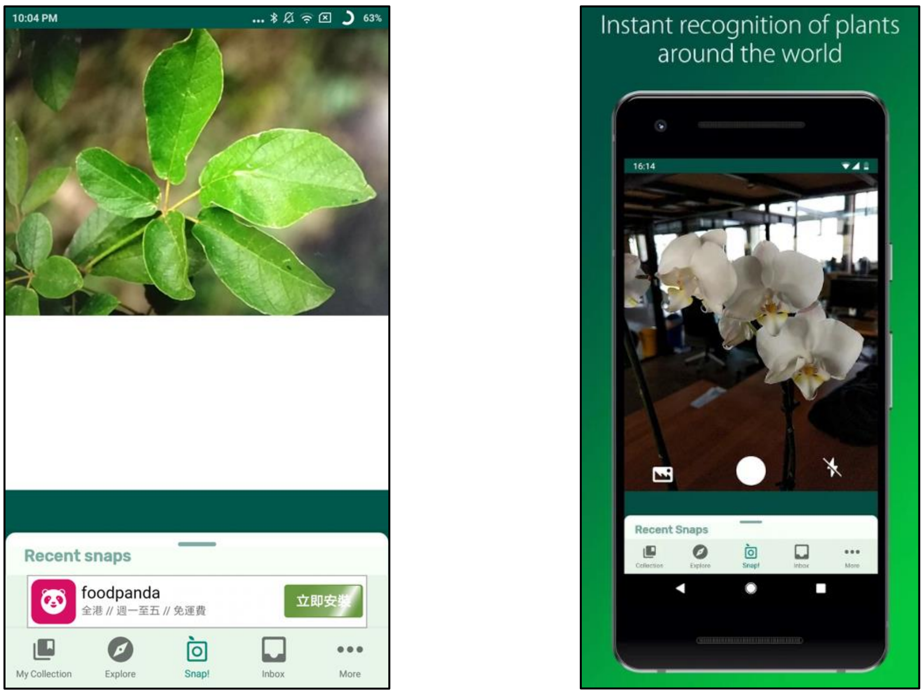


Figure 14 : Failure of capturing picture in PlantSnap and capture in Google Play of the UI

In the application, we cannot use the photo capturing function as there is no button shown to do the task. The problem may be caused by the inadequate testing or not using the common components mentioned as above the compose the photo taking function. We may try to avoid it by call the camera library to finish the photo taking step in our application.

[7] Issue discussed in article "Software engineering issues for mobile application development", the five important are iPhone, Android, BlackBerry, Windows Phone, Symbian. In fact, IOS (15.1%) and Android (84.8%) should be the most significant platform for mobile application, according to the statistical data from IDC on worldwide Smartphone OS.
Reference: https://www.idc.com/promo/smartphone-market-share/os

4.2. Design Specification

The mobile application is developed as the offline AI application to recognize the specie of the plant by the photo of leaves. The following are the specification of the application.

4.2.1. Architecture Design

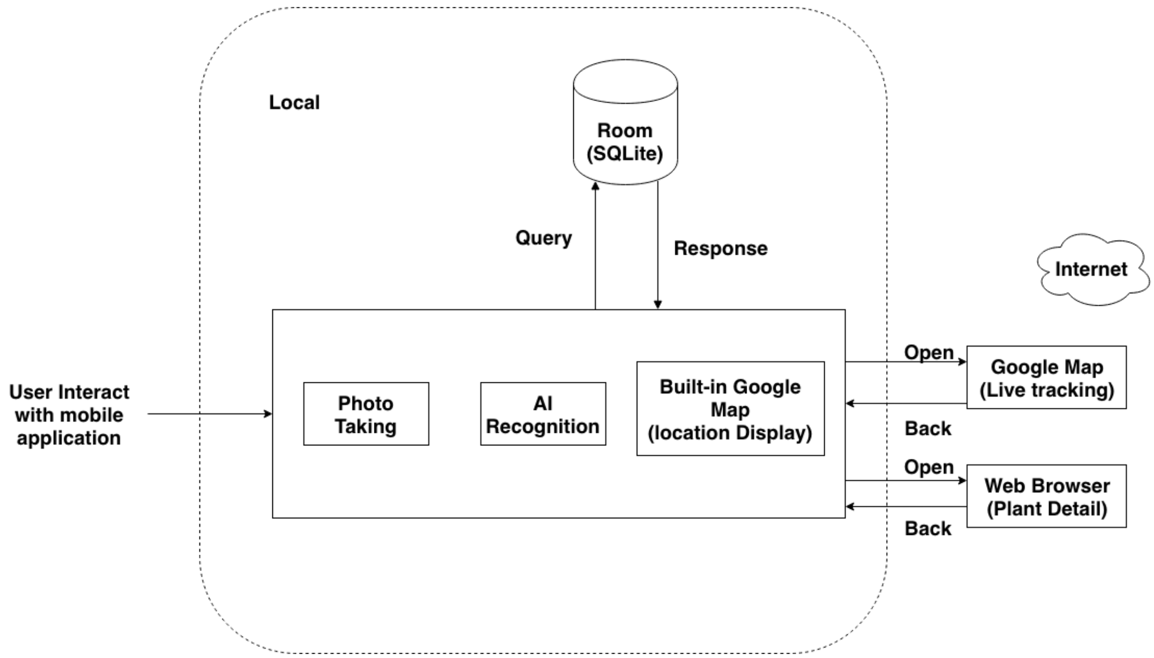


Figure 15 : Architecture Design Diagram

The mobile application is mainly composed by the functions, which can operate independently without internet access. Users will take photo of the leaf of the plant and pass the pictorial data as the input to application. After that, users will gain the plant result by the AI Recognition model locally in the mobile.

If the users want to get the details of the specific plant, they can click the detail button and the application will make an intent, requesting to open the browser and check out the detail of the plant in the website in CUHK Shiu-Ying Hu Herbarium.

If the users want to get the location of the plant in CUHK, they can click the location button and the application will make an intent, to open Google Map showing the particular plant location that we took data at.

The application also provides the built-in Google map that will not require mobile data, and mark all the plants on it that available for user to perform recognition.

4.2.2. Data Flow Diagram

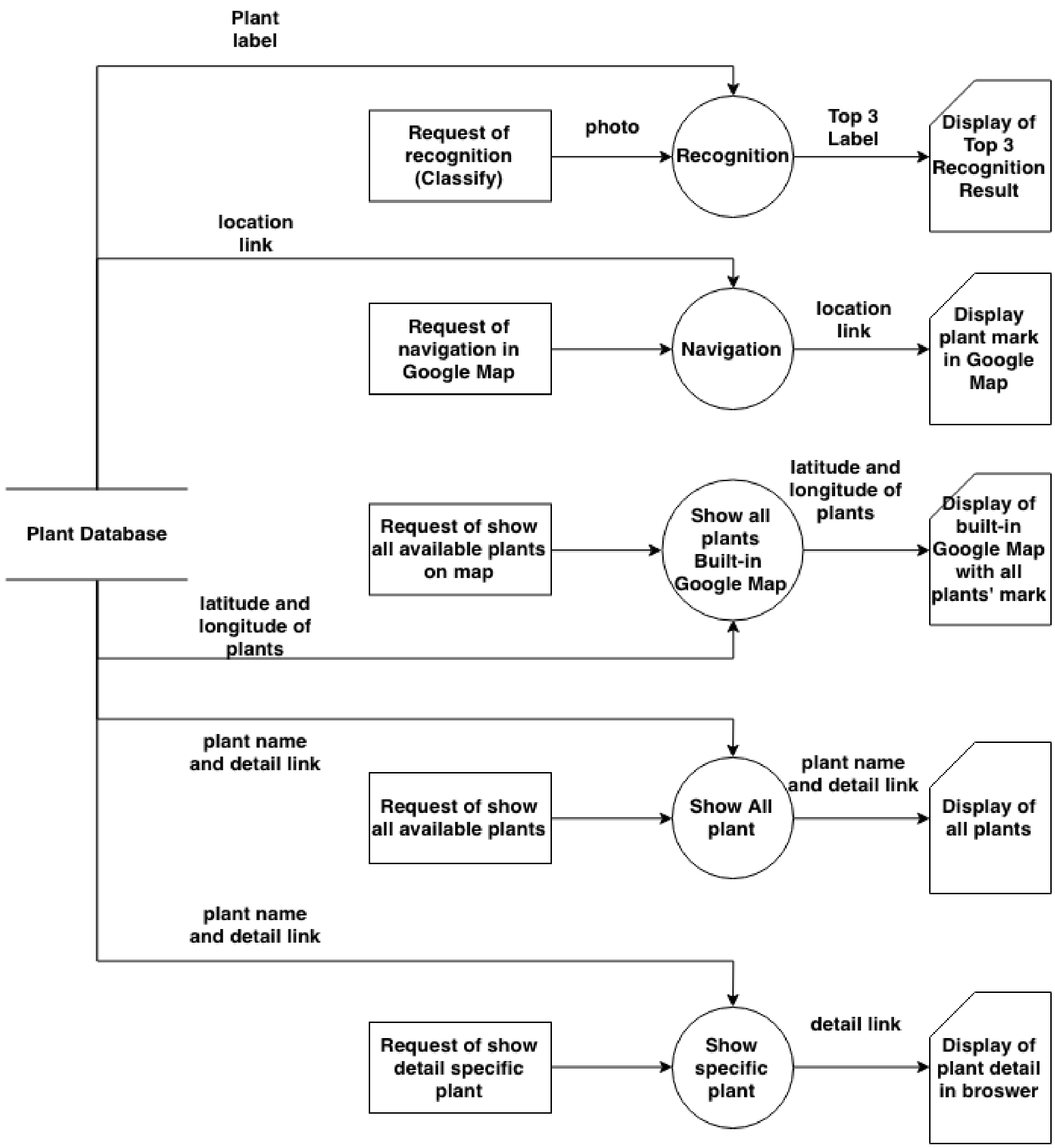


Figure 16 : Data Flow Diagram of mobile application

Five functions in the application share the same database to get their required data in order to implement their function.

1. The "Recognition" function takes the photo as an input, and the model will indicate the top three similar results to the result, only if the score of prediction is greater or equal than 10%.

2. The “Navigation” function takes no input from user, as user click the marking in the built-in Google Map and application will direct users to the Google Map application in the device to do the navigation of the specific plant.

3. The “Show all plants in built-in Google Map” takes no input from user, and get the latitudes and longitudes information of plants, marking their position in the map. The markers are clickable, which means user can access further function in in the Google Map app if they click the mark of the plant.

4. The “Show all plants” function take no input from user and get all the name in Chinese and English of the plants and detail link (collected from CUHK Shiu-Ying Hu Herbarium), and show all plants as CardView[8] to the users.

5. The “Show specific plant” function take no input from user, and it is trigger when the user click the detail button in the activity[9] of recognition result or in the activity of show all plants.

[8] Terminology used in Android Development Document, describing a component showing information, https://developer.android.com/reference/android/support/v7/widget/CardView
[9] Terminology used in Android Development Document, describing a page or a building block in android app, https://developer.android.com/guide/components/activities/

4.2.3. Structural Diagram

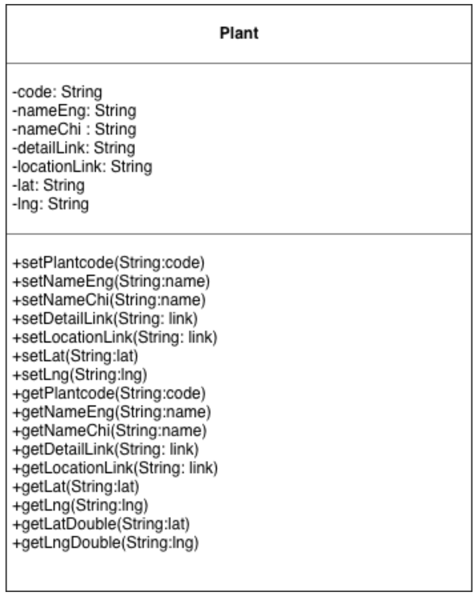


Figure 17 : Table Plant

The application is applied with a simple database with the above table. The code is an unique identifier of the plant, taking the first letter of the scientific name of plant. If two plants have the same first letter of the scientific name, for example “Actaea pachypoda” and “Acer palmatum”, we will add numerical index behind the code according to the time of data collection, like AP1 and AP2.

The name of the plant mainly follows the information provided by the CUHK Shiu-Ying Hu Herbarium, as it is possible to have different naming convention in different place to the same plant.

The lat and lng are representing the latitude and the longitude of the plant we collected in CUHK, there is possible location error because we marked the location with our own mobile device at that moment.

The locationLink is generated by the lat and lng in Google Map, we need the link for requesting the intent of opening Google Map official application.

4.2.4. UMLs

We will try present the application by different UML diagrams, in order to conclude the usage in a more complete way.

4.2.4.1. Use Case Diagram

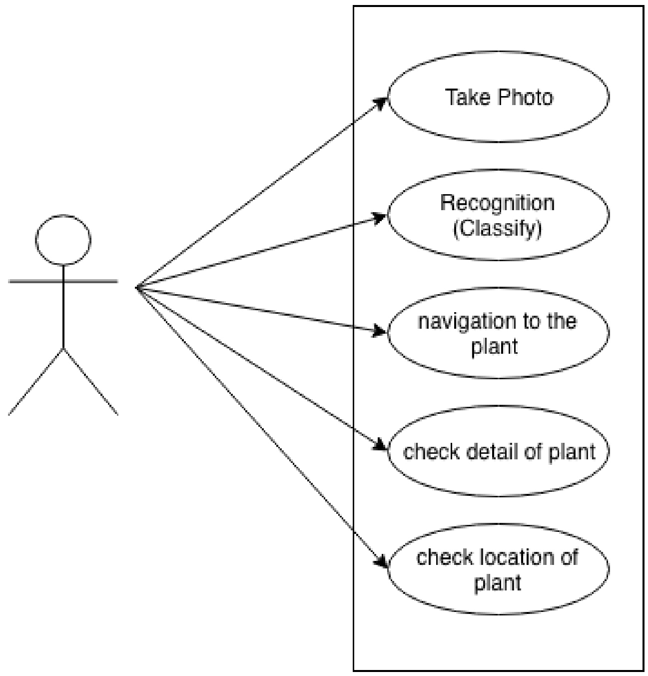


Figure 18 : Use Case Diagram of application

Users can use these five main functions in the mobile application. For “Take Photo” and “Recognition”, they have to implement in the sequential order to get the correct and expected output. They can be performed without internet connection and get the recognition result.

Checking location of plant do not need the internet connection as the marker will be shown in the built-in offline Google Map.

Besides, “navigation to the plant” and “check detail of plant” require the internet connection. If the users want to get the sophisticated information of the recognized plant, they can check it online by pressing the button provided.

4.2.4.1.1. Activity Diagram – Basic Recognition

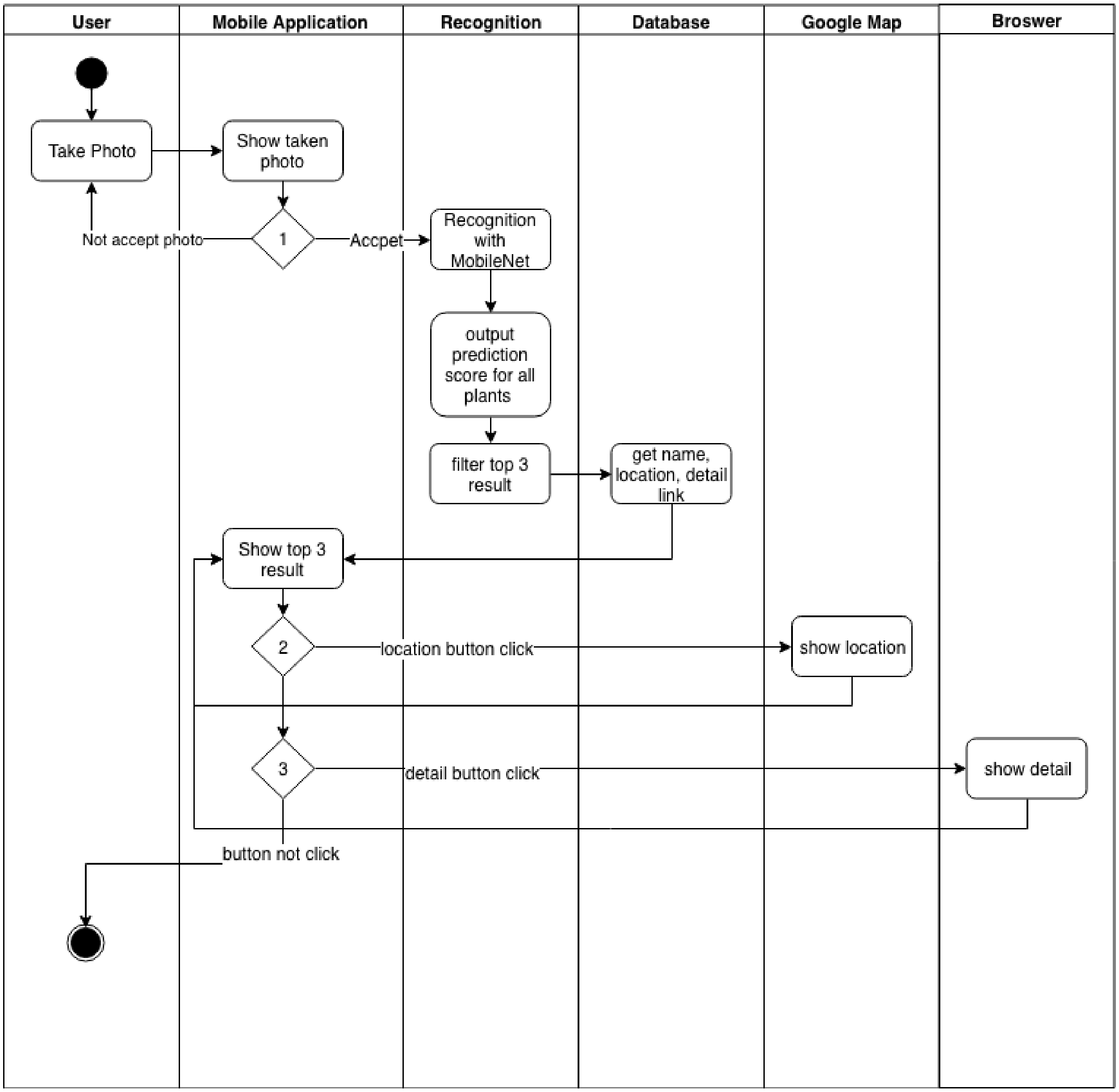


Figure 19 : Activity Diagram of Basic Recognition

Decision block 1: Users can retake the photo for recognition until they satisfy with it.

Decision block 2: It indicates the location button of the plant shown in result activity. If the users click the button, the mobile application will direct the user to Google Map application.

Decision block 3: It indicates the detail button of the plant shown in result activity. If the users click the button, the mobile application will direct the user to browser showing the page of herbarium of the specific plant.

This is the activity flow of a user when they normally use the application to the plant recognition. Up to the database, tasks will perform offline until user click the location button or the detail button of the plant. The activity diagram also shows the modularity of the mobile application, in which if we have to update to database or we would like to change the recognition modal, those amendments will not affect the application and cause bug in system.

4.2.4.1.2. Activity Diagram – Navigation

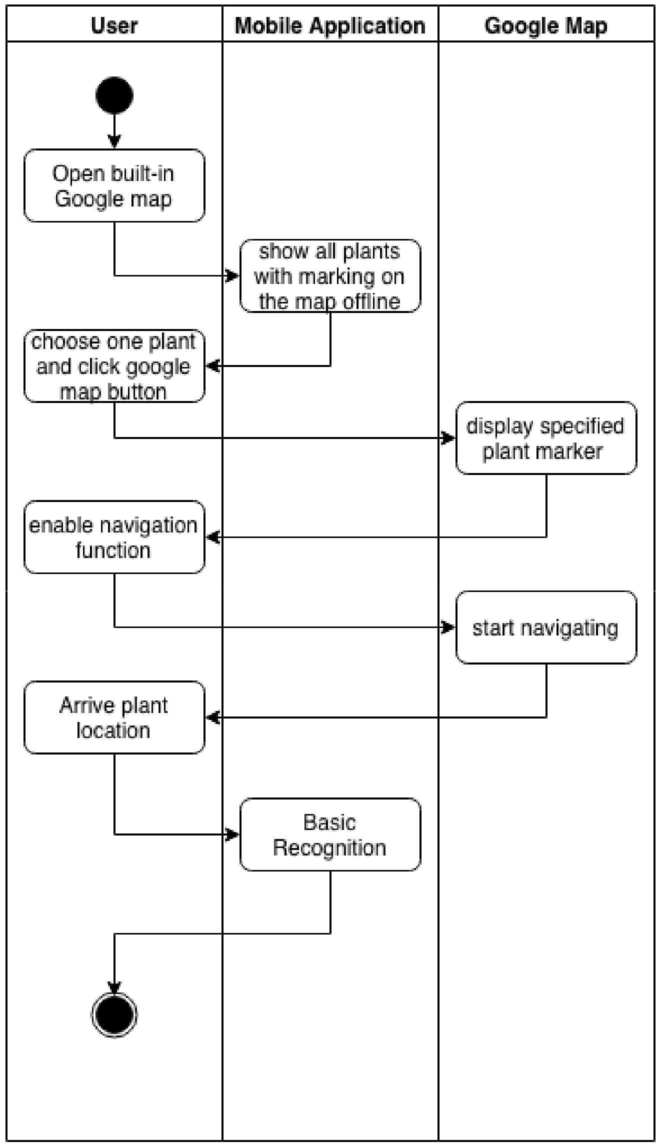


Figure 20 : Activity Diagram of Navigation

This is the activity flow of using the navigation function to reach the plant we record in CUHK. Users will see the location information and marking offline in the built-in Google Map. After user choose a kind of plant he is interested at, he can choose the marker, and click the Google Map direct button to open Official Google Map application. Then user can follow the instruction of navigation to look for the plant. Once users finish navigation, he can go back the main application and try the photo recognition function.

4.2.4.2. Functionality

In summary, user can experience have several functions that the application brings along:

i) Get to know the species of plant by the leaf with photo recognition
ii) Gain the detail of the recognized plant
iii) Gain the location information of the recorded plant in CUHK
iv) Locate and find the recorded plant in CUHK

We hope to deliver an instant plant recognition application without cornering the network issue to the users and help user to acquire the details of the plant if needed.

4.2.5. User Interface

In this semester, we have developed the mobile application of leaf recognition in Android platform, including the function mentioned before. The following will be some capture and procedure of using the mobile application.

4.2.5.1. Screen Capture

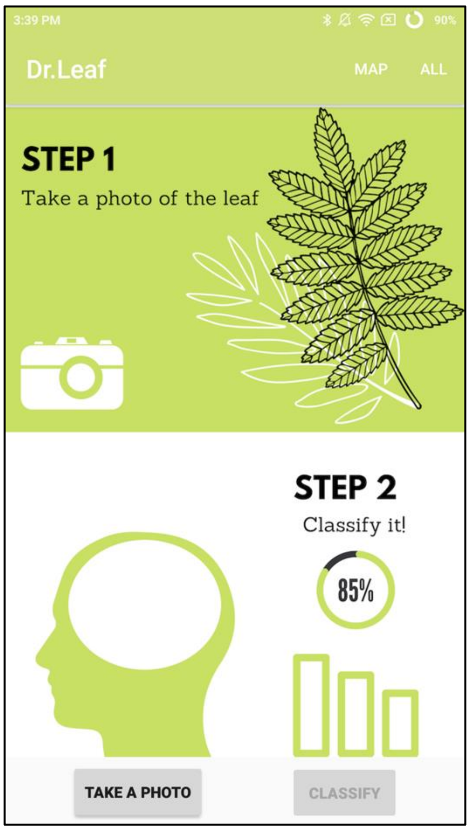


Figure 21 : Main page of Dr. Leaf

Figure 22: Main page after taking photo

Main page provides simple 2-step instruction for user, assisting them to do the recognition. We would like to keep the page as simple as possible, therefore only four buttons appear in the page

Figure 23: Top 3 Recognition Result

Figure 24: Browser Page of plant details

Figure 25: Location mark of the plant in Google Map

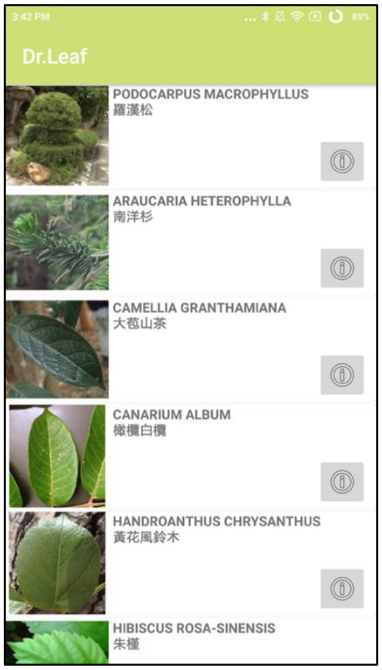


Figure 26 : View all plants in CardView

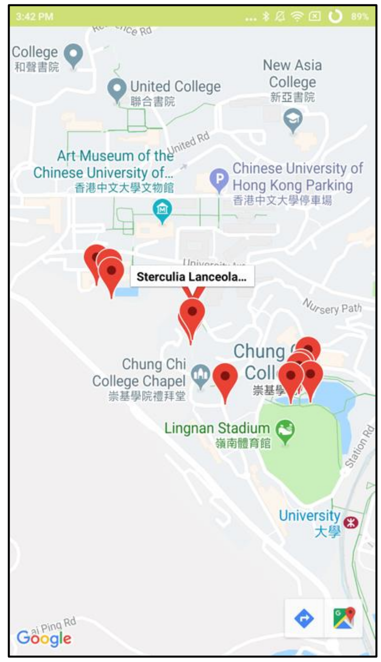


Figure 27: View all plants in Map

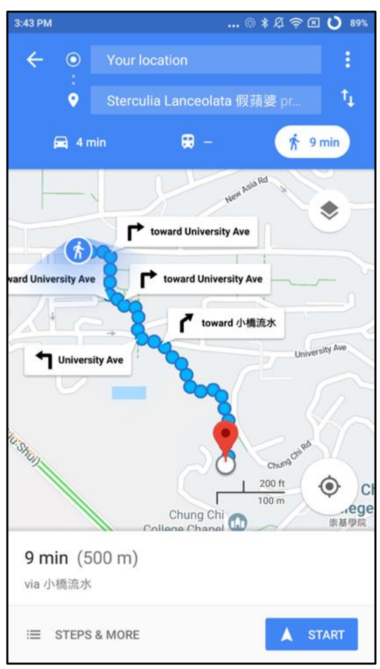


Figure 28 : Navigation in Google Map

4.2.6. Procedure of using Dr.Leaf

4.2.6.1. Basic Recognition

In this function, user can get the recognition result and choose the follow up option they want.

4.2.6.1.1. Step 1

Figure 29 : Main Page of Dr.Leaf

Click the “take a photo button” and shoot the leaf of the plant.

4.2.6.1.2. Step 2

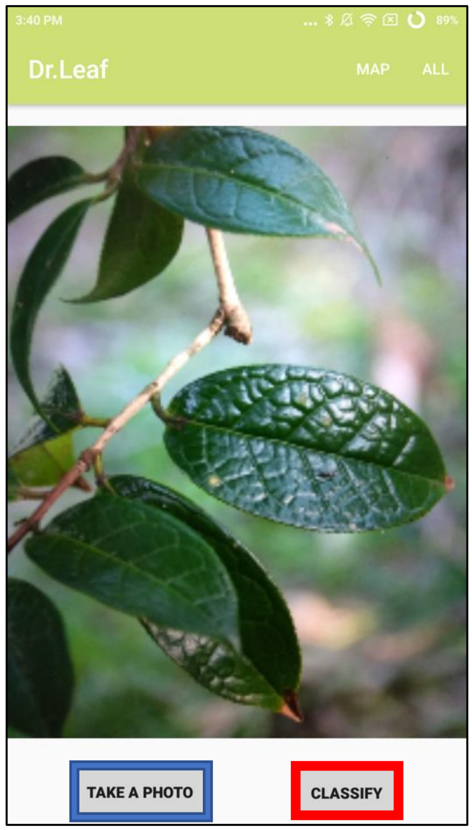


Figure 30 : Taken photo will be displayed

If you are satisfied with the photo, click "CLASSIFY" button to do recognition. (Red box)

Otherwise you can take the photo again by clicking the "TAKE A PHOTO" Button (Blue Box)

4.2.6.1.3. Step 3

Assume the "CLASSIFY" Button has been click with a satisfying leaf photo, Top 3 Result will be shown of the recognition if the score is greater than 10%.

Figure 31 : Recognition Result

Users can choose to see the location of the plant by clicking the location button (Blue Box) or clicking the detail button (Red box) to check the detail of the plant provided by the herbarium.

4.2.6.1.3.1. Step 3.1

Assume user clicked the location button, Dr.Leaf will direct user to Google Map with the location of the selected plant.

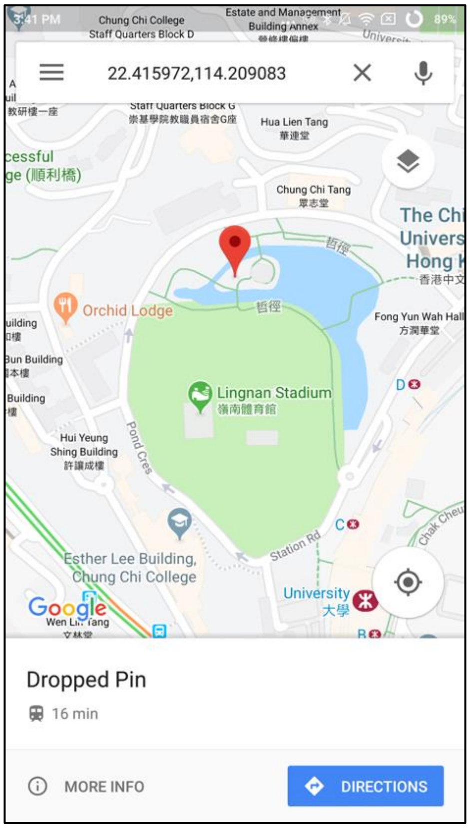


Figure 32 : Google Map with dropped pin

4.2.6.1.3.2. Step 3.2

Assume user clicked the detail button, Dr.Leaf will direct user to the website of herbarium with the location of the selected plant.

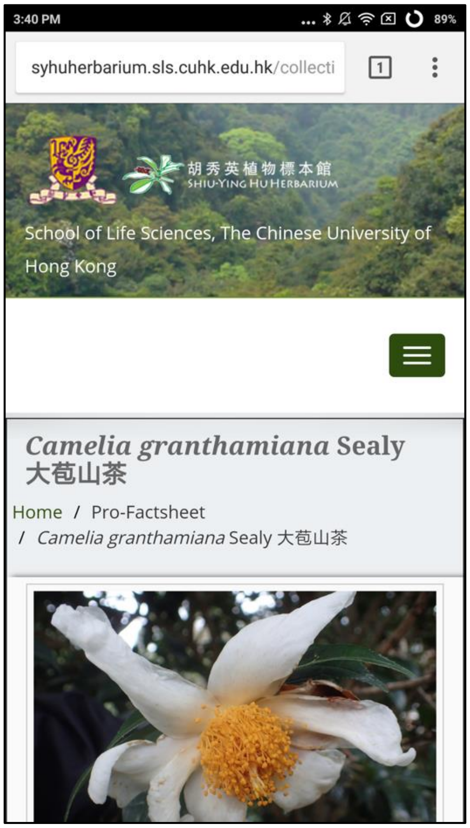


Figure 33 : Herbarium Website showing the Plant with tag "ca"

4.2.6.2. View All plants and navigation

User can check out all the recorded plant in application.

4.6.1.2.1. Step 1

Figure 34 : Main Page of Dr. Leaf

There are two ways to view all the plants. User can choose to view all the plants in built-in Google Map (Blue Box) or view all the plants in list (Red Box)

4.2.6.2.1.1. Step 1.1

Assume user chose to view in list.

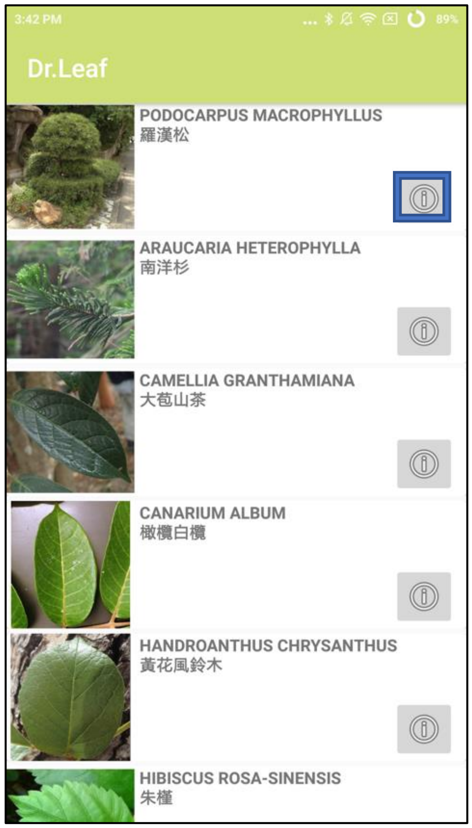


Figure 35 : Show all plants in list

User can click the detail button to get the detail of the plant (Blue box).

4.2.5.2.2. Step 2

Assume user chose to view in map.

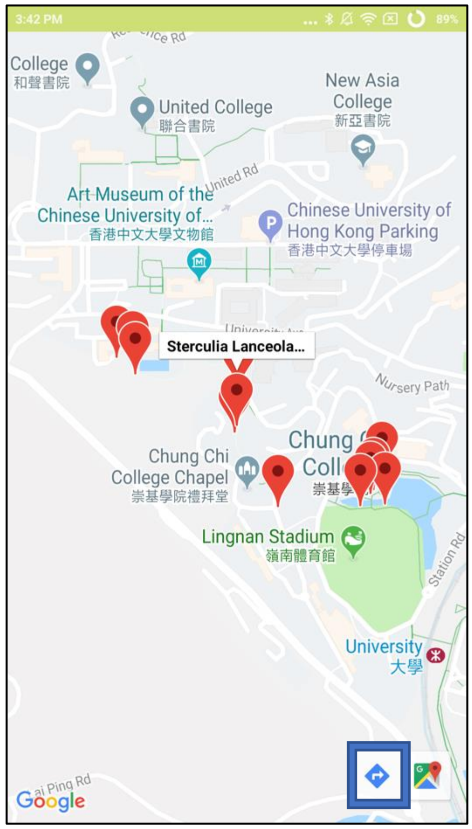


Figure 36 : View all in Google Map

User can view all the plants as dropped pin with their name. If user click the navigation button (Blue box), Dr.Leaf will direct him to the Google Map and do so.

4.3. Testing and Evaluation of Dr. Leaf V1.0

4.3.1. Database

Dr.Leaf has an offline database in the local device, which does not allow user to insert, update, or delete the data in the dataset. As we are using the API provide by Room[10], we can control every task or query to the database in the developer level and user has no right to access the database unless they do the recognition and query the plant indirectly.

4.3.2. Logical Statement

In the source code of the application, we seldom use logical statement to process data flowing in the application. There three two kind of data always flowing in the application.

i) Photo, pictorial data

ii) Query result which is in the format of String

iii) Intent[11] request

The first two of data are directly being passed in the application, for example, passing the photo to the classifier and passing back the recognition in String. The third data is mainly concerned by the android system, regarding the process that has to been run for the next step of the application. It is also being passed in the application without logical decision. Therefore, testing on logical statement is not required in the application.

4.3.3. UI Testing

In the first prototype of the application, we discovered the application works perfectly fine in the emulator but fail to open in the actual mobile device of the pages that contain picture. After checking the coding error and potential mistake made by developer, we realized the picture, for instance, the icon of the plant and the instruction in the main page has a large size, consuming large amount of resource in an intent. When the memory is large enough and exceed the limitation of an intend[12], therefore we reduce at least 40% of the size of the images and the application finally well fine in the mobile device.

[10] Google develop and provide the Room as an abstraction layer of SQLite, we can easily handle the database by Room, https://developer.android.com/training/data-storage/room/

[11] Terminology used in Android Development Document, describing an operation in android app, https://developer.android.com/reference/android/content/Intent

[12] It is mentioned API 23 is 517716 bytes (~ 0.5mb) from https://www.neotechsoftware.com/blog/android-intent-size-limit

5. Project Details (Specification of Dr. Leaf V2.0)

5.1. Overview

In phase one, the product of this project is a plant recognition mobile app, which makes use of the mobile net neural net and leaf image to differentiate the species of the plant. A basic structure of the mobile app is implemented. It provides the function of recognizing ten kinds of plants and provides the information of those plants.

Recall that the goal of this project is to build a platform for the public to collect leaf images and the AI model could be optimized. In phase two, an auto-training module is added to the mobile app. The module is divided into two parts, which are the app-sided and the server-sided program. The mobile app provides users an interface to contribute the leaf data. On the other hand, the server provides computation power for re-train of the AI model.

Users can use the plant recognition function in country side since internet access is not necessary. However, if users want to help improving the classification quality, users may need to connect their device to the internet for contributing their image data to the dataset. With a larger dataset, the trained AI model would usually perform better.

5.2. Data Transfer Method

To allow users help improving the classifying power, data transfer between the mobile app and the server is needed. The mobile application needs transferring new data to the server. On the other hand, the server needs transferring back the trained AI model to the mobile app.

In this project, Http method would be used to transfer data. Inside the mobile application, OkHttp, a network library, would be used to implement the function of sending the Http requests. Inside the server-sided machine, Django web framework would be used to build a server for handling Http requests.

5.3. Mobile Application

In phase two, several changes are applied to the Dr. Leaf V1.0, including the addition of the data contribution interface and model update function, and other adaptive changes. In the following sections, the details of changes will be discussed, and those that are not changed would not be covered to avoid repetitive description.

5.3.1. Architecture Design

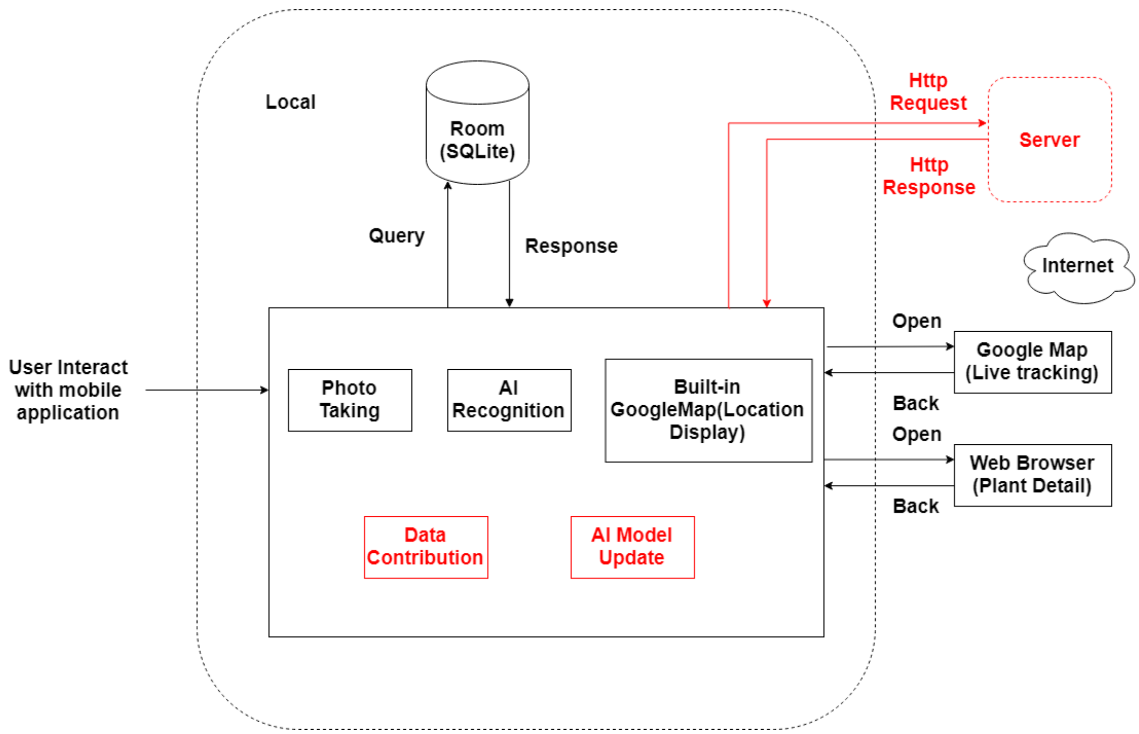


Figure 37App-sided Architecture Design Diagram (those in red are added in phase two)

The mobile app consists of the following function:

1. AI recognition
2. location display
3. species information providing
4. data contribution
5. model update.

As shown in fig, those in red are newly added to the project in phase two, which are data contribution and AI model update. These two functions together with the model training server compose the auto-training module. They play the role respectively in each part of the machine learning workflow "data collecting and then model training". The module is separated into app-side and server-side due to the limited storage and computing power.

5.3.2. Data Flow Diagram

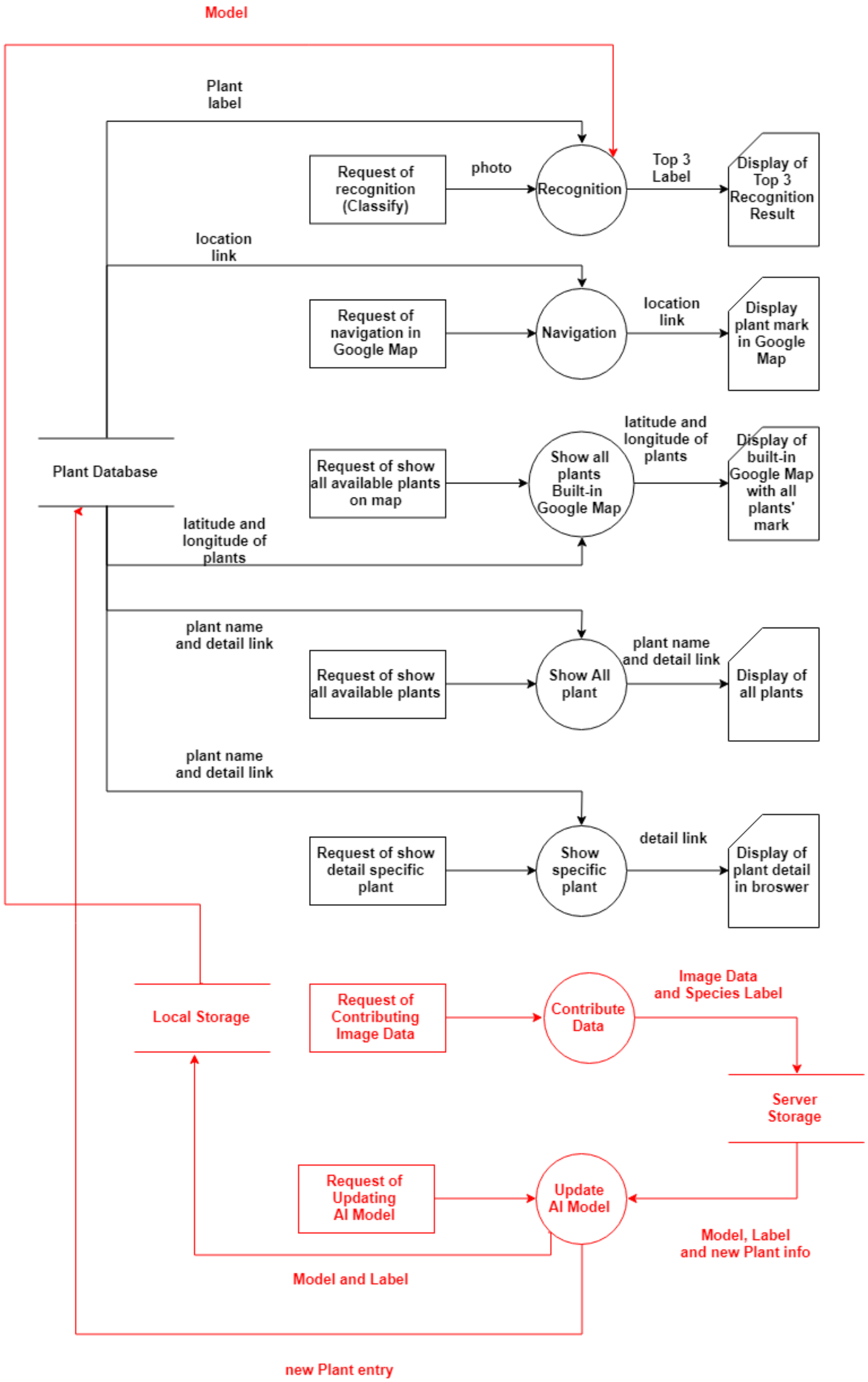


Figure 38Data Flow Diagram of mobile application (those in red are added in phase two)

The two newly added functions are served as the interface between app and server. Data between the app-sided storage and the server-sided storage would flow through them.

The “contribute data” function can be triggered by users after the object is classified. It relies on the user input to get the Chinese and English species name. The image data together with the species name will be sent to the server for further processing.

The “model update” function fetches the trained model, label and new species information. The model and the label are stored into the local storage for later classifying the plants. The new species information is changed to a new plant entry. The plant entry is then added to the database.

5.3.2. UMLs

We will try present the application by different UML diagrams, in order to conclude the usage in a more complete way.

5.3.2.1. Use Case Diagram

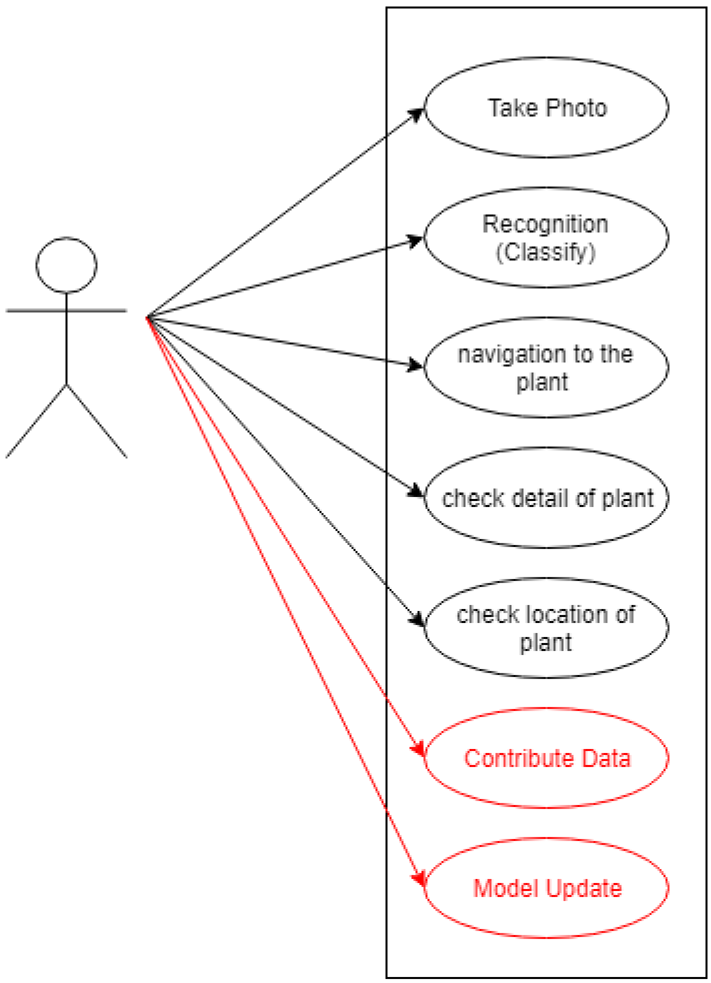


Figure 39Use Case Diagram of application (those in red are added in phase two)

Besides the five original functions, two are added in phase two (those in red in above figure).

“Contribute Data” allows users contribute the correct labelling of the image, when they found that the classifying result is not satisfied. It requires internet access to send the data to the server.

“Model Update” allows users fetch the up-to-date AI model back to local storage. It also requires internet access to communicate with the server. One should note that the server will not generate a new model only after new data are contributed to the server.

5.3.2.2. Activity Diagram – Model Update

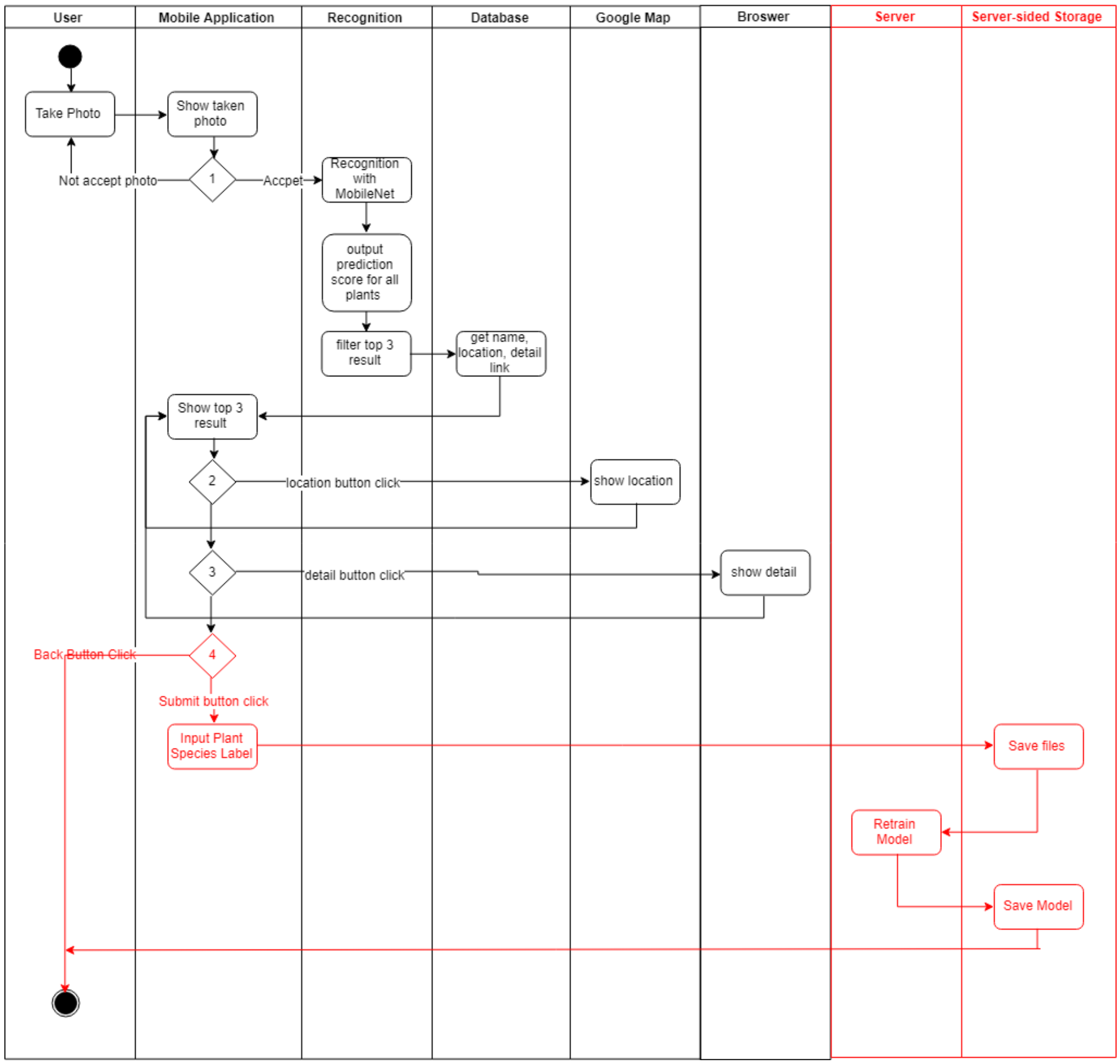


Figure 40Activity Diagram of Basic Recognition and Data Contribution (those in red are added in phase two)

This is the activity flow of recognizing the leaf images and contribution of the image together with the species label. Users is able to contribute their leaf images after the recognizing process. For the detail activity flow description of the recognition part, please read to previous section.

Decision block 4: If users want to contribute data to improve the recognizing model, they can tap the contribution button. If they tap the back button, the system brings users back to the main activity.

After users entered the contribution activity, users need to input the Chinese and English species name. The corresponding data are sent to the server. The server then re-train the AI model and bring back the control to the users. Users will be brought back to the main activity.

5.3.2.3. Activity Diagram – Model Update

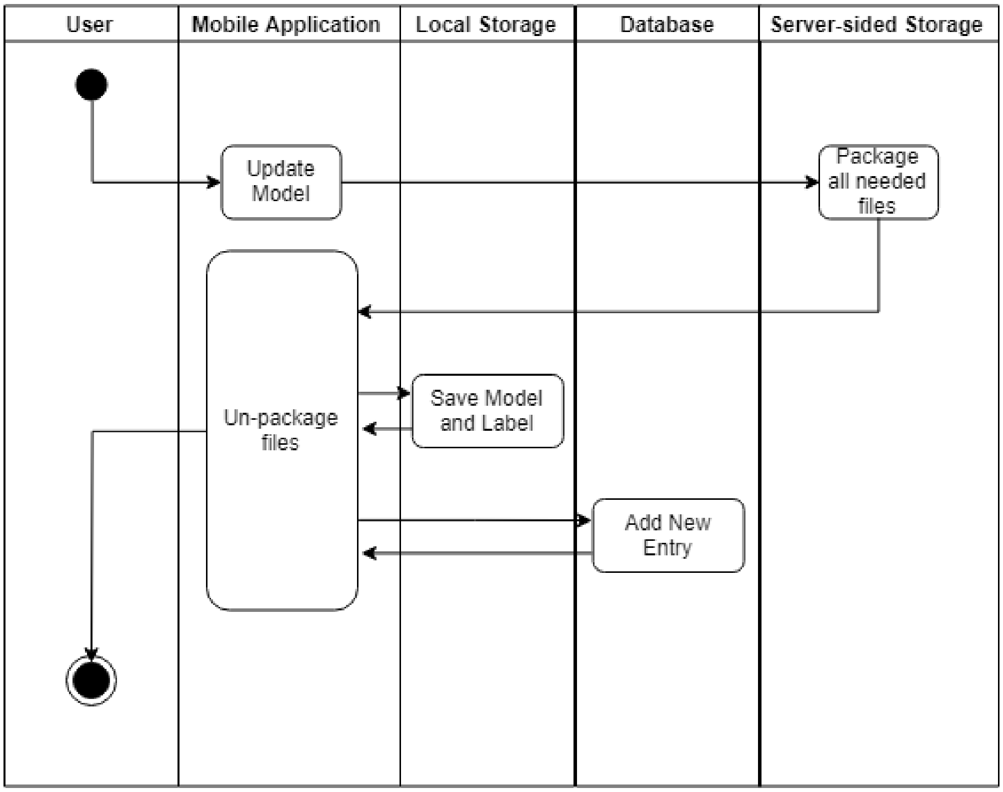


Figure 41Activity Diagram of Model Update

When the users want to update model, a http get request is sent to ask server to prepare required data. The server will package the needed files and send the package as a response. Mobile app receives and un-package the files. It then saves model and label to the local storage. New plant entries are also added to the database of the mobile app. After storing the model, label and plant information, the whole model update process is finished. Users can use the new model to recognize the plants.

5.3.2.4. Functionality

In summary, users can help improving the recognizing model through adding new data of different plant species. Besides contributing data and update of the model, users can use the instant plant recognition application anywhere without cornering the network issue.

5.3.2.5. User Interface

In this semester, we have updated the Android mobile application of leaf recognition. The following are some captures and procedure of using the mobile application. Those components that are not modified in phase two would not be repeatedly mentioned. Please refer to previous sections.

5.3.2.5.1. Updated Layout

In main page, a new "update" button is added on the menu bar for users to update the newest model when their devices are connected to the internet.

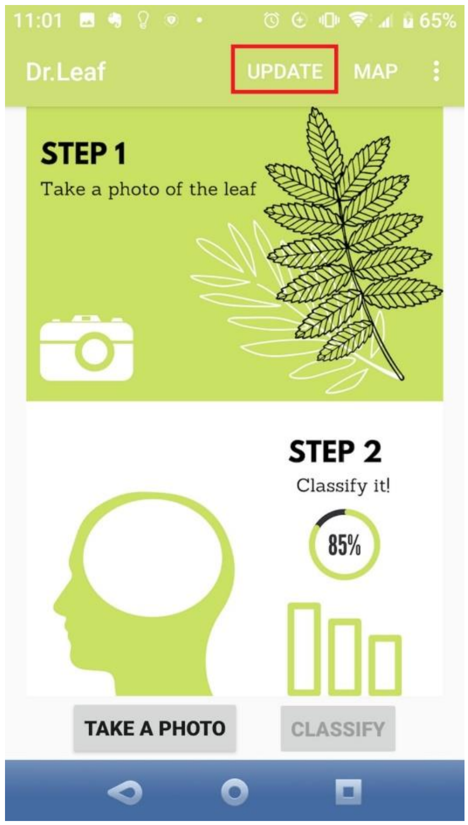


Figure 42Update the model

In phase one, the herbarium webpage is provided for detail information of the plant in CUHK. After phase one, the webpage information is no longer provided. A google search of specific kind of plant is given to the users instead.

In phase one, data in this page are almost hard-coded. In phase two, new plant species can be added to this page by users' contribution.

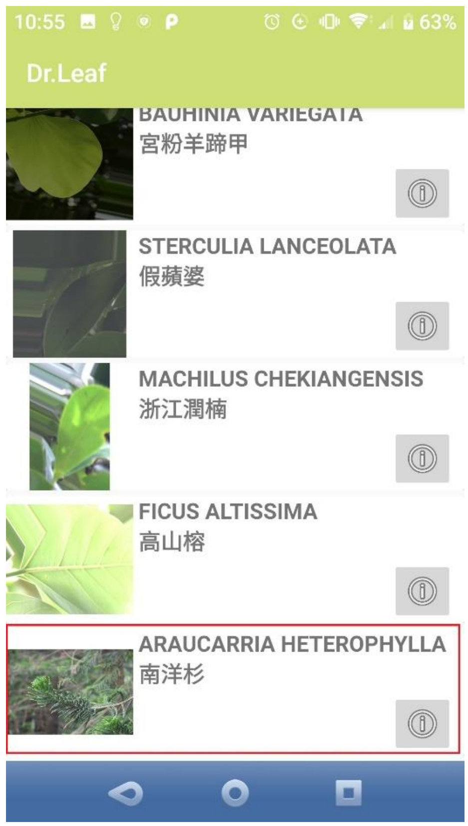


Figure 43New plant entry is added

5.3.2.6. Procedure of using Dr.Leaf

5.3.2.6.1. Data Contribution

After the classification, users can contribute the correct species label together with the leaf image if they are not satisfied with the classifying result. Besides correcting the label into the existing ones, users have the option to add new plant species into the database. Allowing this can increase the extensibility of this plant recognizing tool.

The following are the screen capture of the workflow:

1. Taking a photo

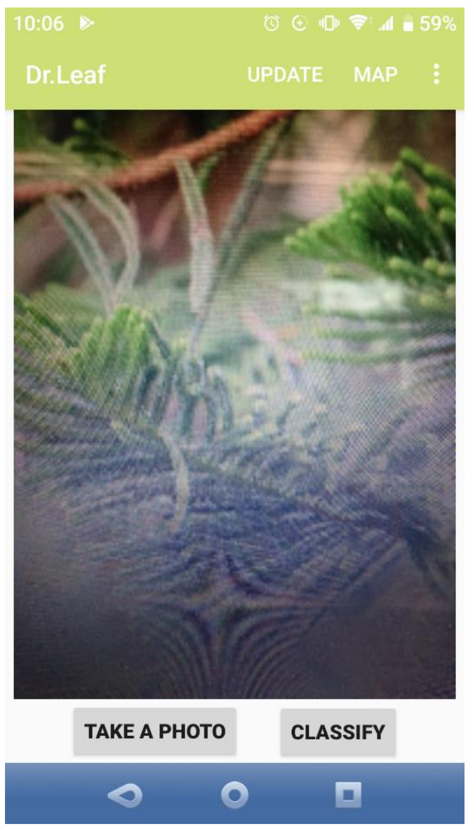


Figure 44Take a photo

2. Check the classification result

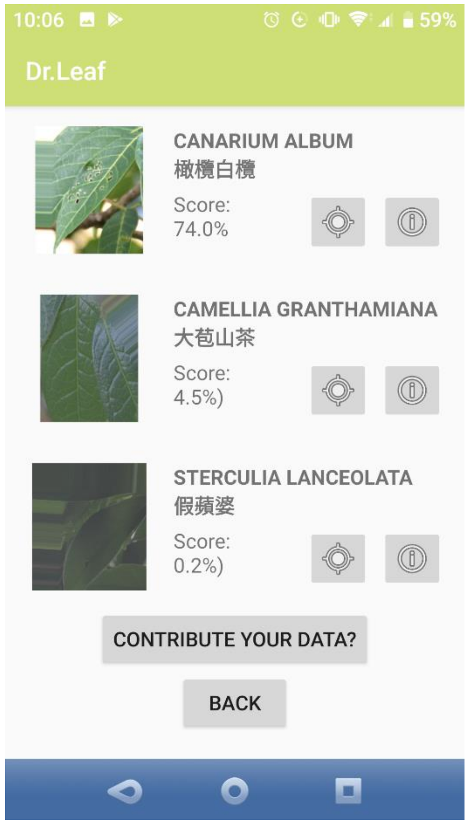


Figure 45Classification result page

3. Contribute the data with label

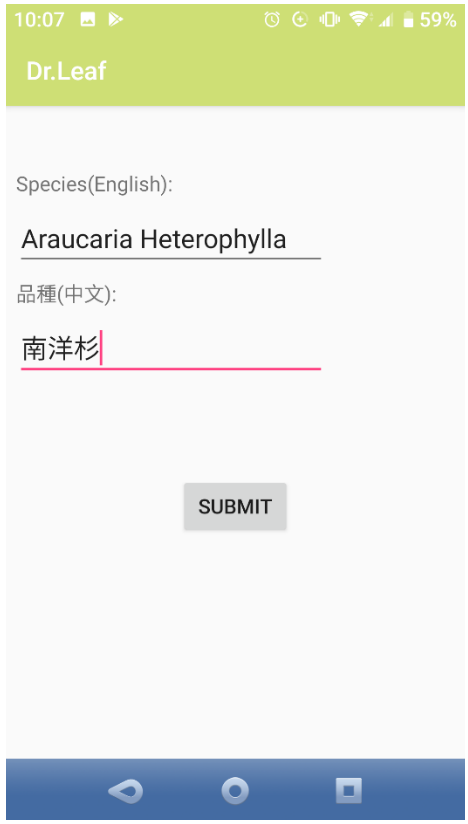


Figure 46Data contribution page

5.3.2.6.2. Model Update

The goal of this project is to build a platform for users to contribute the leaf image data. When the number of users is large, the model could be updated from time to time. Users have the freedom to update their model at a suitable time.

Clicking the update button and wait for several second, the model would be updated.

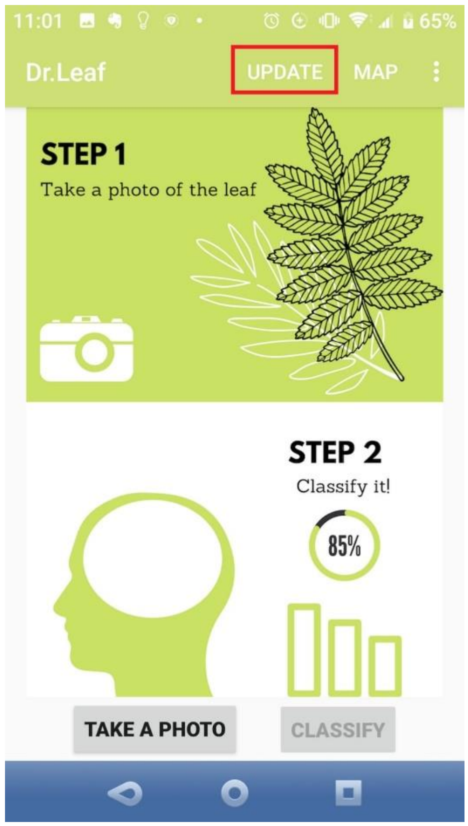


Figure 47Update the model

5.3.2.6.3. Model Validation

After update of the AI model, users may want to validate if there is any improvement. Users can perform the classification process again on the same object. Then users can compare the results.

The following are the screen capture of the workflow:

1. Take a photo

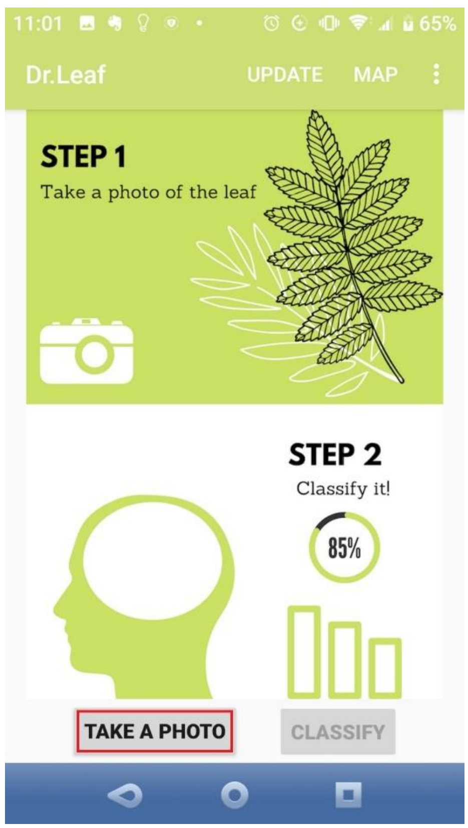


Figure 48Take a photo

2. Adjust the photo

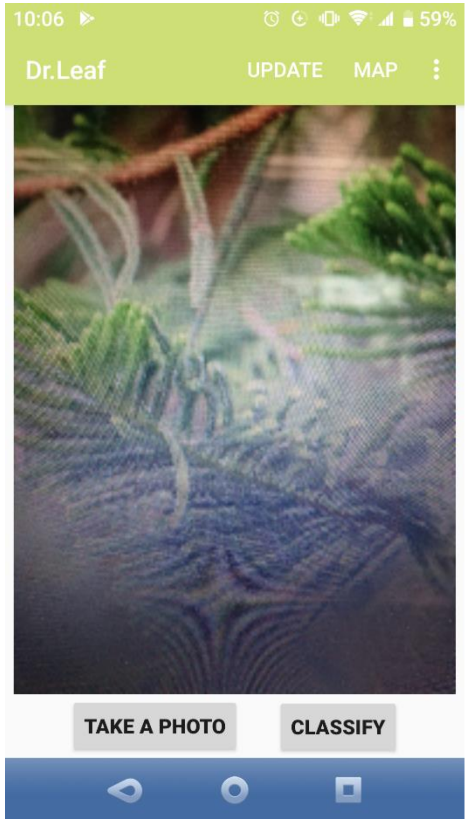


Figure 49Adjust the photo

3. Check the classification result

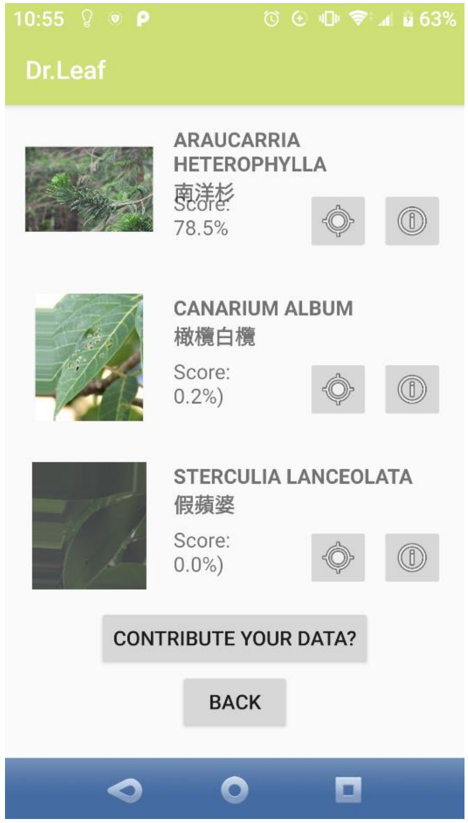


Figure 50Update on Showing all leaves

5.4. Server

The goal of this project is to develop a powerful plant recognizing tool. This tool relies on the state-of-the-art technology neural network. To obtain a better neural network model, a larger and high-quality dataset is always needed. Collecting data is a long-termed and exhausting work. Therefore, focus on building a large and sustainable platform is a good idea on obtaining data from users' contribution.

Inside the application, a "data retrieval, model re-train, and model update" workflow is needed. However, with the limited storage and computing power of mobile device, storing a large dataset and re-training the AI model is nearly impossible. Therefore, rely on remote machines with large storage and satisfied computing power is necessary.

In the following section, the detail in consideration on hardware and software of server-sided solution will be discussed.

5.4.1. Hardware

5.4.1.1. The Needs

As mentioned above, extra computer resources are needed for storing the dataset and re-training the AI model.

A decent machine is not the only requirement. As our goal is to establish a platform, we could predict that the demand of storage and computing resources would be increased continuously. The whole system should be easily extended.

5.4.1.2. Consideration on Hardware

Last few years, cloud computing became a hot topic. More and more enterprises make use of cloud computing in order to control the cost of maintaining the hardware. However, cloud computing is not always the best fit to everyone. In some case, private hardware may be more suitable.

5.4.1.2.1. Safety Issues

One of the concerns about cloud is the storage safety. Most cloud providers provide only public cloud. That means storage of different users is only separated logically under cloud providers' control but not physically separated into different hard-disks. When malicious users found vulnerability of the cloud system, they may make use of it to steal data from other users besides exploit the network vulnerability.

Besides, as a company, cloud providers would not expose the internal detail of the cloud. Therefore, we would never know whether the cloud providers would monitor users' activity, or even steal users' data. Therefore, for those who needs to handle sensitive and important data, cloud may not be suitable.

5.4.1.2.2. Learning Curve

There may be less safety issues on private hardware. However, maintaining the private hardware could be a burden to users. Setting up the hardware requires users having special knowledge. For example, since we need scalability in our project, we need to build an intranet for connecting the computers and database. Learning these kind of special knowledge needs extra efforts comparing to using cloud directly.

Conversely, in order to gain the market share in the cloud computing market, most cloud provider would simplify the cloud platform interface, provide beginner fee discount and put resources into beginner tutorials to make learning their platforms more comfortable.

5.4.1.2.3. Flexibility

For most of users, the demand of hardware changes from time to time. For example, the target users of this project are mainly CUHK students and they would be less possible to use the tool at night. In this case, idle machine over the whole night is a waste. On the other hand, we could predict that the size of the dataset would be increased continuously. Adding of extra storage could be easier in cloud. The whole process can be done programmatically in short time.

In terms of flexibility, cloud resources can be managed easily to fulfil users' requirement. Therefore, significant waste on idle resources and insufficient resources are less possible to be observed on cloud platform.

5.4.1.3. Choice of Hardware

For consideration on learning curve and flexibility, cloud computing would probably be a good choice to this project. Learning how to setup a machine could be easy and would probably be finished within one hour. After the setup of a virtual machine and establishment of the interface, using the virtual machine would be as easy as using a personal computer. On the other hand, we only need to pay for the resources we need rather than waste money when the machine is idle. And we could easily scale up in order to cater to the needs of storing the continuous-growing dataset.

In terms of data security, data leakage could be a problem. However, this project is a non-profit project so the data leakage would not cause any financial loss. The advantage of using cloud computing clearly overweight the risk of data leakage.

Because of the huge benefit of using cloud and our confidence built from that many big enterprises also transferred their business to the cloud, using virtual machine from the cloud providers could be a good choice.

5.4.1.4. Choices of Cloud Platform

As a small-sized initial project, there is not much requirement on the cloud platform. What we want is only a stable and scalable machine. Therefore, there is no special requirement on the cloud platform. The well-known Amazon Web Service is used.

5.4.2. Django Web Framework

Besides the hardware, the server software is more important to server. Django Web Framework is a python web framework which enables rapid web development. Considered that there are abundant high-quality python library and the python community is big, as a web development beginner, we can easily get support from the python community. Therefore, the Django Web frame would be used to build the server-sided software in this project.

5.4.3. URI Design

As mentioned in previous section, the communication method between the server and mobile app is using HTTP protocol. The URI would be treated as an API to the server resources.

5.4.3.1. Data Contribution

User can contribute data through HTTP post request on the following URI:

http://ip_address:port _number/images

5.4.3.2. Model Update

User can update the model through a HTTP get request on the following URI:

http://ip_address:port _number/models

5.4.4. Architecture

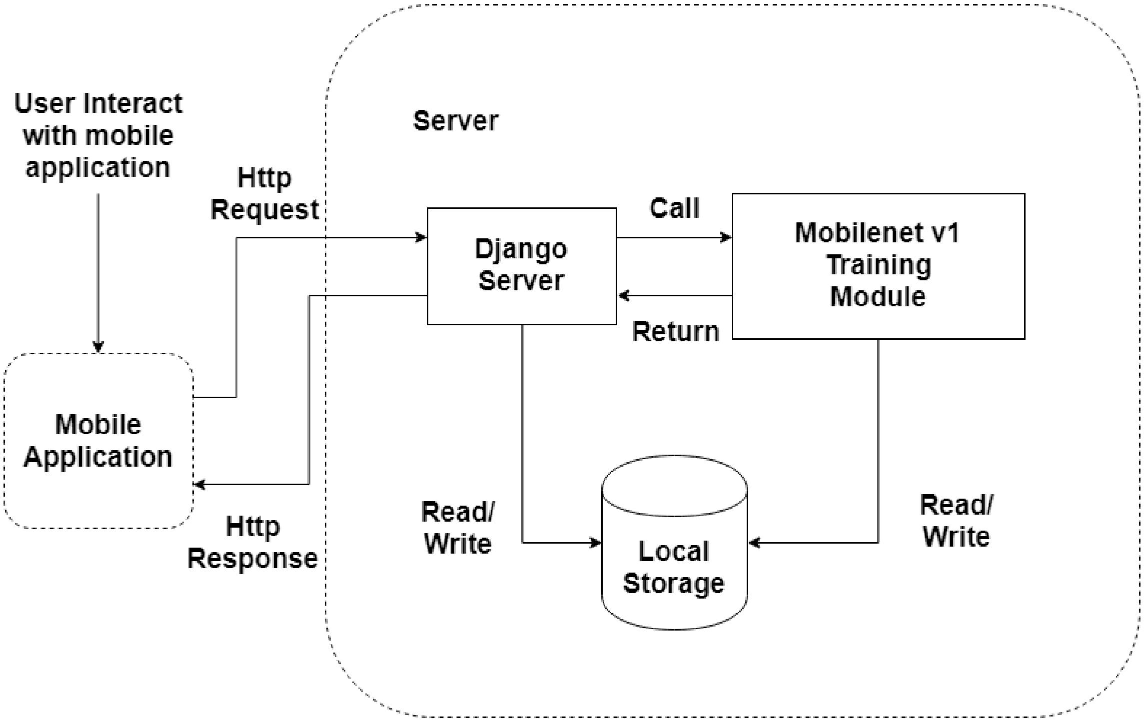


Figure 51Server Architecture

The server-side environment is mainly composed of the Django server and the mobile net v1 module. The local storage is used to store useful files, including the dataset, model, label and plant species information. The mobile net module is used to re-train the plant recognizing model. The Django server is served as an interface between the other components of the server and the mobile application. It is used to handle the incoming http request from the mobile app. According to the request, the Django server would either retain the model or send needed files back to the mobile app client side.

5.5. Development Aspect

5.5.1. Bottom-up development

In the development of both mobile app and the server, the bottom independent modules are developed first. After checking the correctness of all bottom modules, the modules would be composed to further check the correctness.

5.5.2. Debug on Dependent Module

For those modules that are dependent on other modules, a simulator would be built for debugging purpose.

For example, the data receiving function on the server is dependent on the request sending function on the app. As shown in the figure below, a request sending HTML interface is made to simulate the app-sided request sending function.

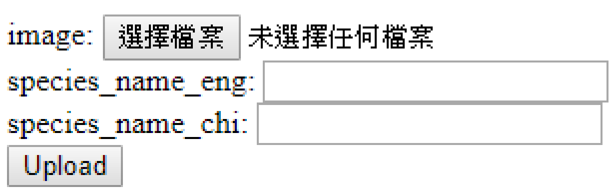


Figure 52request sending simulator

Using this approach allow us debug on a single module each time. If there is any error, the error could be spotted easily.

6. AI Model

6.1. Background

After AlphaGo, a computer program designed to play Go utilizing neural network, beat the best Go player in the world, more and more people heard the name of neural network. Neural Network is a popular machine learning model with great power in finding the relationship between input and output. By finding the relationship between decisions on playing chess and winning probability, AlphaGo showed the power of neural network on playing chess. By adding convolutional layers, neural networks can also do good job in image recognition. We believe that this tool should be handful to our tree recognition projects.

6.2. Problem Defining and Solution

To recognize trees, we are solving a classification indeed. For some elements drawn from different sets, we want to find out which set they belong to. Usually same set of elements share some common features. We can compare the difference of features of different elements to find out which sets the elements belong to. For example, we can differentiate apple trees from banana trees with the color of their fruits.

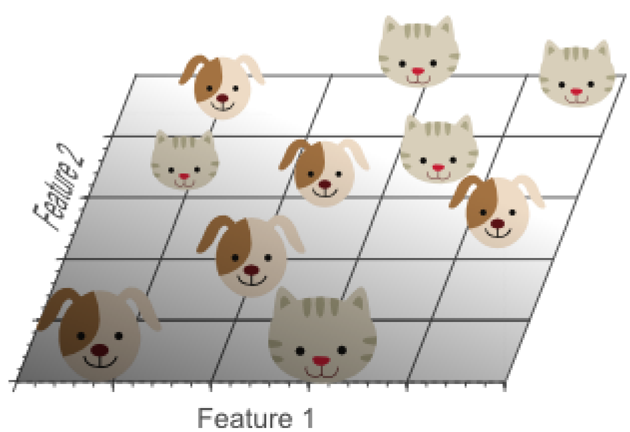


Figure 53: Classification between cats and dogs using 2 features.[13]

[13] The image is obtained from http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/

6.3. Neural Network

Traditionally, people select features by experience to solve classification problem. Eventually, the performance can be unstable. By simply feeding the features into the model, neural networks have the power on selecting useful features for classification. Therefore, we can avoid the unstable performance by make use of neural networks.

Literally, Neural Network is a network of processing units called neuron. Neuron is an activation function of the weighted sum of input. Inside the neural network, the weights are the parameter. We should train the model using optimization algorithm like gradient descent in order to get the suitable parameters.

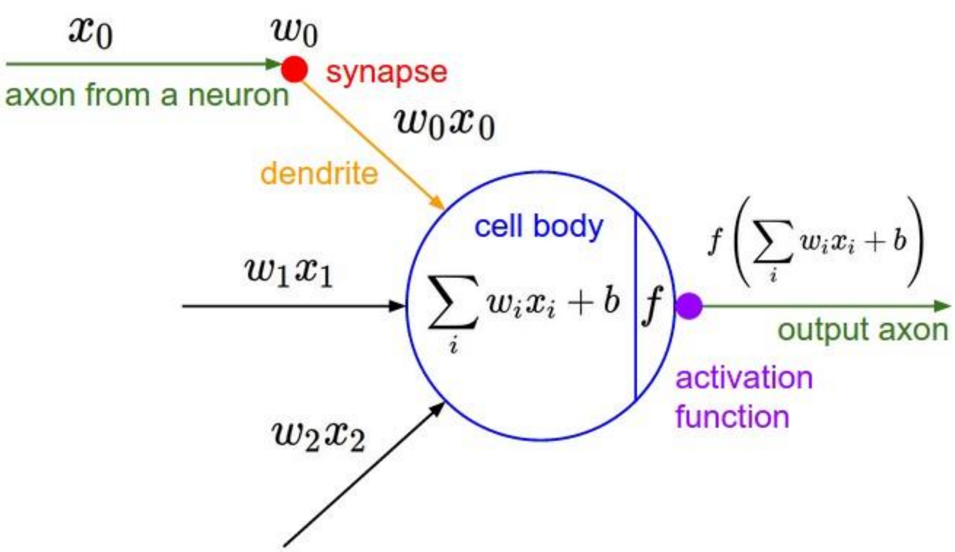


Figure 54: A neuron in neural network[14]

For different purpose, the neural network architectures can be different. The most commonly used ones are feed-forward network and recurrent network. For simplicity, people normally use feed-forward network. It has been mathematically proofed that

[14] The image is obtained from https://stackoverflow.com/questions/38850538/single-neuron-in-neural-network-using-c

standard multilayer feedforward networks can approximate any measurable function to any desired degree of accuracy.[15] In reality, the performance of neural networks varies but we can we can overcome this problem with the increase in depth of neural network.[16]

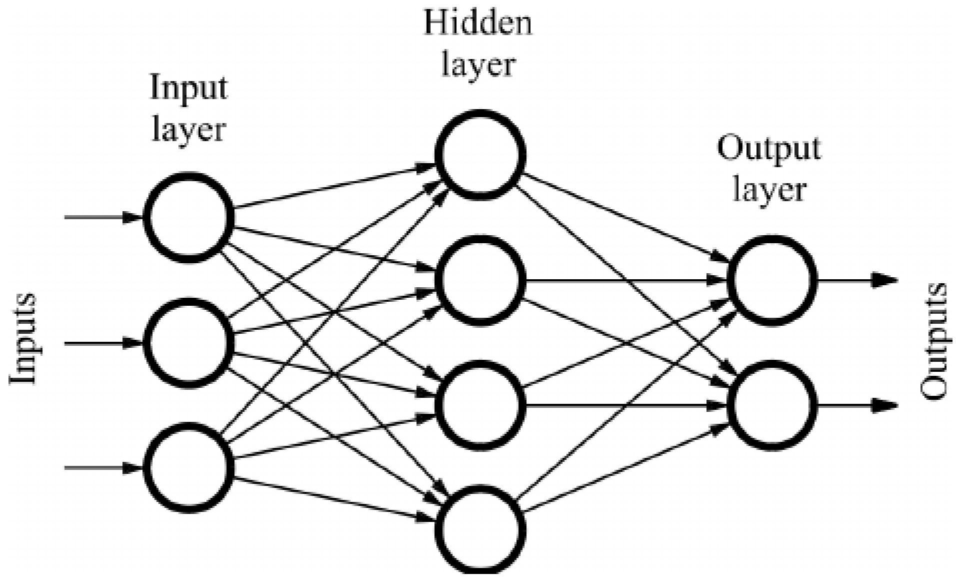


Figure 55: A feed forward neural network[17]

6.4. CNN

As mentioned before, the architecture of neural networks can varies depend on the problems and the needs.

Starting from 2012, the appearing of Alexnet, a Convolutional Neural Network (CNN), leads to a great success in image classification with the help of strong calculation power of modern graph processing unit. Using convolutional layer, Alexnet obtained top-1 accuracy of at least 60% on several ILSVRC dataset and became the champion of the ImageNet Large Scale Visual Recognition Challenge in 2012, which is a world-class competition.

[15] The proof is in the article “Multilayer feedforward networks are universal approximators”.

[16] This is discussed in the article “Very deep convolutional networks for large-scale image recognition” and “Going deeper with convolutions”.

[17] The image is obtained from https://www.researchgate.net/figure/Sample-of-a-feed-forward-neural-network_fig1_234055177

CNN is a kind of neural network in which convolutional layers are added. In the convolutional layer, convolution is performed for the input and features can be extracted from the images easily. To get useful features, we have to tune the parameters in the convolution kernel by training the model.

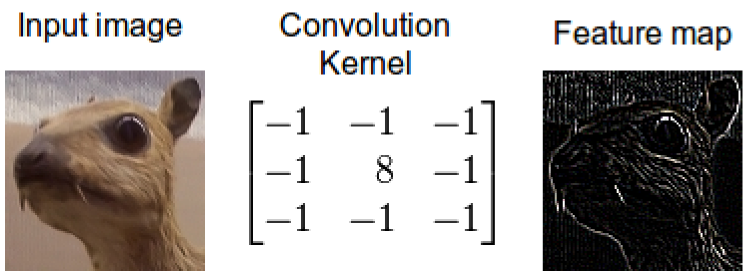


Figure 56: Output image(right) was obtained by performing convolution operation using the convolution kernel(middle) on the input image(left).[18]

In early years, neural network is seen as a black box because it is difficult to understand what is happening inside the network. Recent years, researcher developed some visualization approaches to investigate the properties of neural networks. In figure 43, features extracted from different layers are shown. One can easily observed that, patterns extracted from the layers that is near the input are relatively simple, such as line and dot. And patterns extracted from the layers that is near the output are more complicated. We can inference that CNN first extracts different features, then combines the simple features to complicated features and finally using the abstract features to classify objects.

[18] The image is obtained from https://developer.nvidia.com/discover/convolution

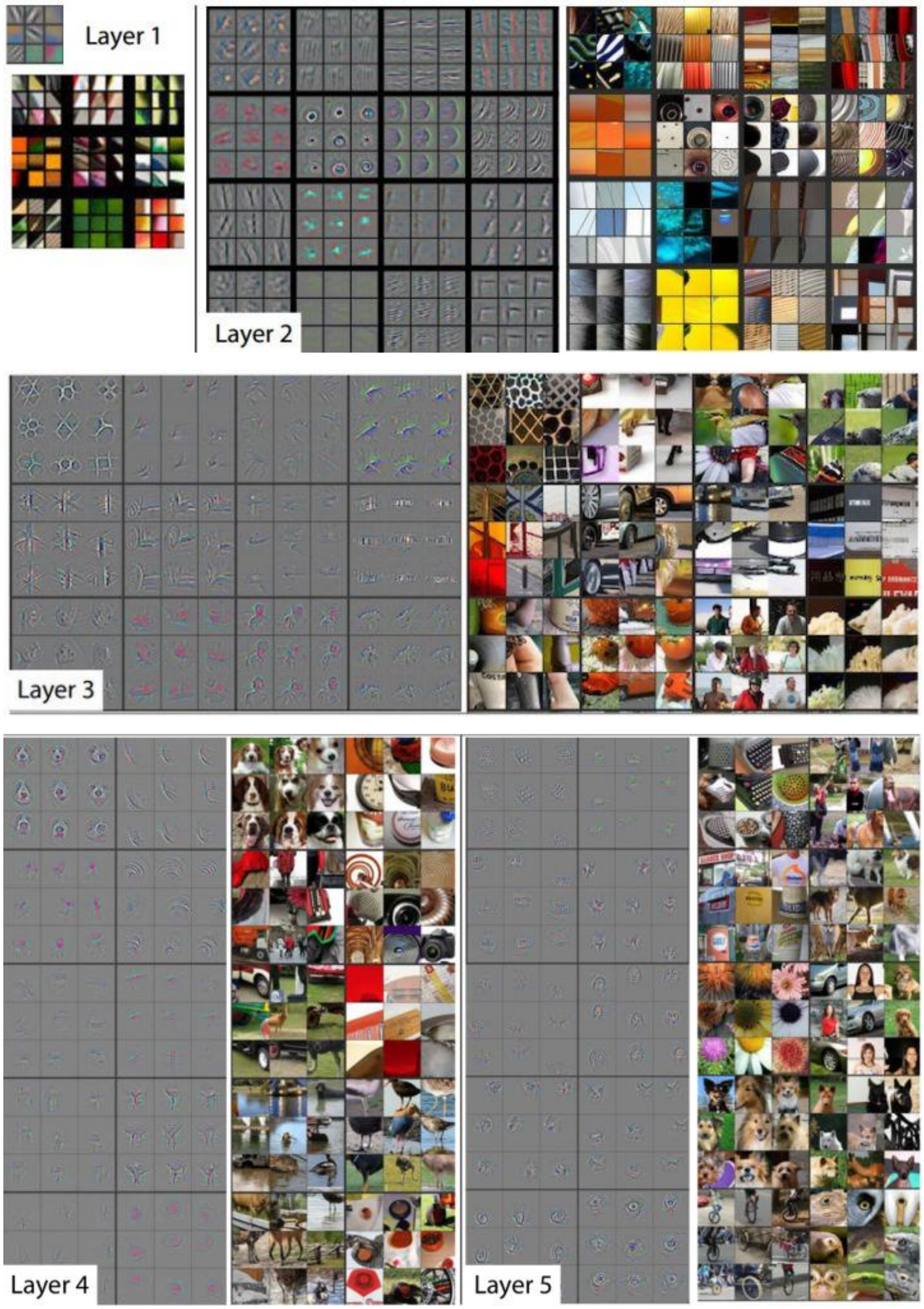


Figure 57: Visualization using deconvolutional approach. The figure shows features extracted in different layers.[19]

[19] The image is obtained from Visualizing and Understanding Convolutional Networks

6.4.1. Concern

CNN is a powerful tool in classifying objects so we have no reason not choosing it. Since the success of Alexnet, CNN developed rapidly and a large amount of CNN variant were invented. Before choosing the suitable one, we should examine the requirements of our project.

6.4.2. Portability

The expected result of this project is to develop an offline mobile application. This means all the calculations would happen in the mobile device. The model should be small enough so that it can be stored in the device.
To persuade the customers to use the application, we should consider their needs. The size of the product should be reasonable. If the application occupies a large portion of storage, the user will not be willing to install it.

6.4.3. Speed

As an application, slow processing could downgrade the user experience seriously. Slow applications are impractical and could probably lose the market. The reasonable processing time should be within a few seconds. We should make a balance between speed and other aspects.

6.4.. Develop Period

As a starting point, the application can only classify ten species of trees in this phase. In the future, we expect we would increase the number of trees the application can recognize. If the training time is too long, it is difficult to do testing and update the application immediately.

6.5. MobileNets

After considering the above concern, MobileNets seems to be a suitable candidate to our project. MobileNets is a CNN architecture develop by Google and features in small size and high speed. The more exciting thing is that the decrease in model size and the increase in running speed would not cause a huge decline in accuracy.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
5× Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Figure 58: MobileNet V1 architecture[20]

[20] The table of the architecture is obtained from the article "Mobilenets: Efficient convolutional neural networks for mobile vision applications".

6.5.1. Depth-wise Separable Convolution

The difference between MobileNet V1 and normal CNN is the adding of depth-wise separable convolution. Depth-wise Separable Convolution is a factorized convolution that has extremely small size with similar accuracy when comparing with some popular models. Traditional convolution layer has the function of extracting features and combining features. Depth-wise Separable Convolution separates them and consists of two parts: the depth-wise convolutional layer which is responsible for the feature extraction, and the pointwise convolutional layer which is responsible for combining features.

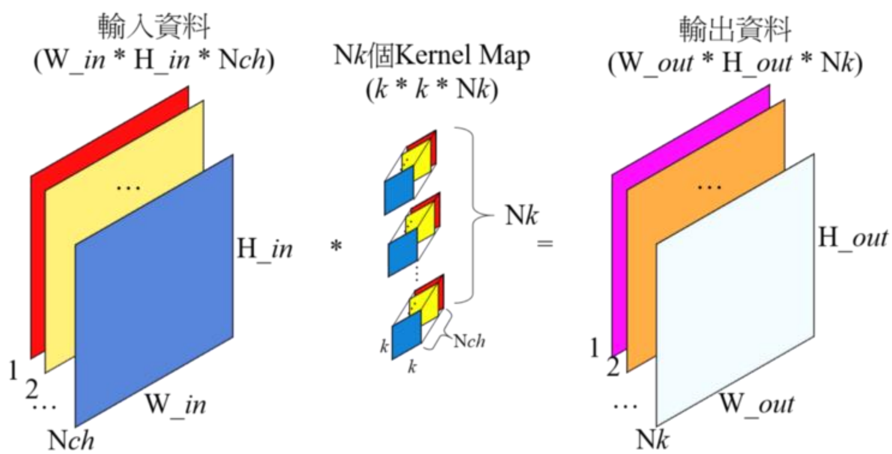


Figure 59: Normal convolutional layer(The symbol '' is the convolution operation)*[21]

[21] The Picture source: https://medium.com/@chih.sheng.huang821/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-mobilenet-depthwise-separable-convolution-f1ed016b3467

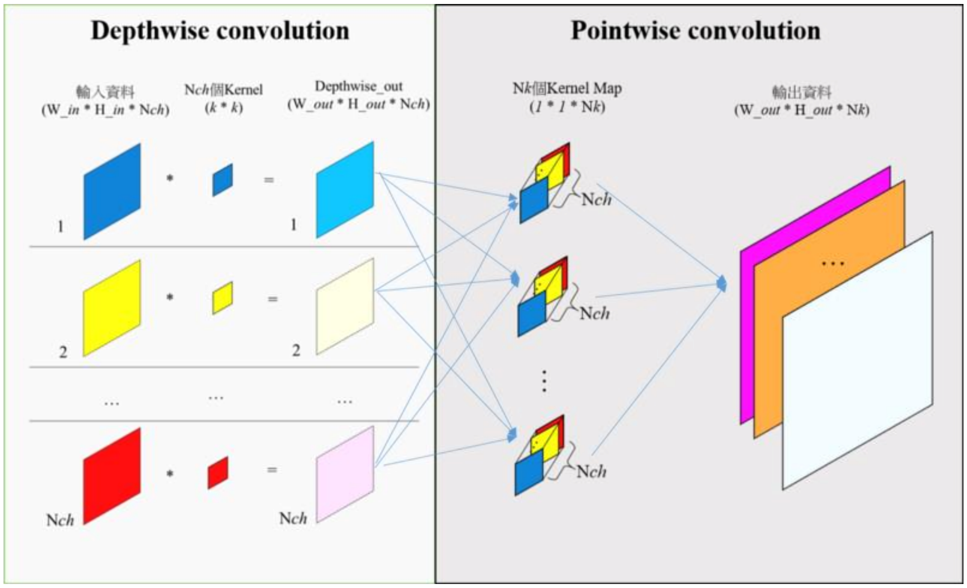


Figure 60: Depth-wise Separable Convolution (The symbol '' is the convolution operation)*[22]

6.5.2. Hyperparameter

Beside the special design of convolutional layer, MobileNet V1 provides two hyperparameters, which are the resolution multiplier and the width multiplier, for controlling model size and computation speed.

Resolution multiplier control the input image resolution. Lower resolution of input image means less pixels, the amount of calculation can be reduced and therefore the training time and testing time can be reduced.

Width multiplier control the size of convolutional kernels. A small width multiplier can reduce the size of the model. Also, the decrease in number of parameters means less calculation and the speed would increase.

However, reduction in resolution would lead to the loss of features. Therefore, the model might not be able to extract enough features and the accuracy would be lower. And smaller kernel size would downgrade the ability of the model.

---

[22] The picture is retrieved from https://medium.com/@chih.sheng.huang821/%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-mobilenet-depthwise-separable-convolution-f1ed016b3467

There is no magic in this two hyperparameter but MobileNet V1 gives us a choice to make a trade-off between accuracy, and size and speed. Figure 47 and 48 shows the trade-off between accuracy, speed and size.

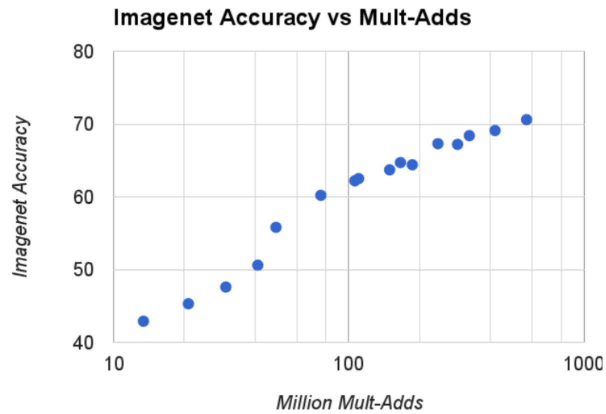


Figure 61: Trade-off between accuracy and speed in terms of number of mult-add operation[23]

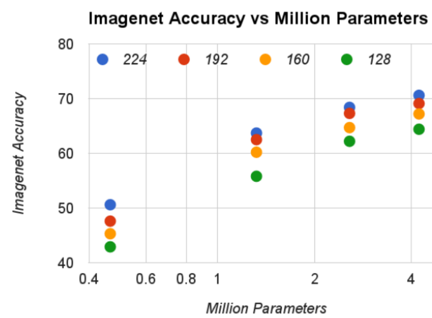


Figure 62: Trade-off between accuracy and size in terms of number of parameters[24]

[23] The figure is obtained from the article "Mobilenets: Efficient convolutional neural networks for mobile vision applications".
[24] The figure is obtained from the article "Mobilenets: Efficient convolutional neural networks for mobile vision applications".

6.5.3. Performance

From figure 49, if a 3*3 convolution kernel is used, the depth-wise separable convolution uses about 9 times less calculation than standard convolutions since output length N in hidden layers are small and can be omitted.[25]

From figure 50, with a special design, the number of parameters of depth-wise separable convolution is less than other popular CNN architectures. Although depth-wise separable convolution and standard convolution is not equivalent, depth-wise separable convolution gives similar accuracy in practice when comparing with other popular CNN architecture.

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2}$$

Figure 63: Ratio of calculation amount between depth-wise separable convolution and standard convolution. Numerator: Amount of calculation of depth-wise separable convolution. Denominator: Amount of calculation of standard convolution. D_K: Length of convolution kernel. M: Number of channels of input image. N: Number of channels of output image. D_F: Length of input image. [26]

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Figure 64: Comparison on performance between standard MobileNet and some popular CNN architectures. First column shows the accuracy of different architecture. Second column compares the speed in terms of million mult-adds. Third column compares the size of differ.[27]

[25] The discussion of the amount of calculation is in the article “Mobilenets: Efficient convolutional neural networks for mobile vision applications”.

[26] The formula is obtained from the article “Mobilenets: Efficient convolutional neural networks for mobile vision applications”.

[27] The table is obtained from the article “Mobilenets: Efficient convolutional neural networks for mobile vision applications”.

7. Experiment and Investigation

In this section, different experiments will be carried out in order to improve the accuracy of recognition and make an attempt to explain the neural network. We would like to the find the solution and answer of how to maximize the accuracy of leaf recognition.

7.1. Experiment I: Data Augmentation

In real application, the actual classification might be affected by the optical distortion caused by the camera, in which the input image to the classifier can be result in a wide range of variation. For example, if user hold the mobile and take the photo close to the leaf, barrel distortion[28] will be appeared in the image [1]. Although such kind of distortion could be corrected by the software calculation in the camera application, other issue like different users can take photo of the same leaf in a random perspective. Therefore, another wide range of variation exists.

In order to solve the above problems, we would like to apply the method of data augmentation. Not only can we generate more data under the situation of limited resource, but we can also simulate the fact of different angle, different distance of photo taking, and different positions of taking photo in the real application. In the below experiment, we would compare the difference of score of prediction with two model, one is retrained with the original dataset, another one is retrained with the augmented dataset. After that, we will upload and install the application in the mobile and take photo of real leaves and record the score for 10 trials.

[28] Different kinds of distortion such as Optical Distortion, Pincushion Distortion, and Mustache Distortion will be caused based on the situation of taking photo.

7.1.1. Specification of Augmentation

In the augmentation, every origin photo went through the augmentations around **5** times under the following conditions[29],

i) No ZCA whitening
ii) Rotation within 40 degrees with respect to the origin of the photo randomly
iii) Width shifting 20% of the total width
iv) Height shifting 20% of the total height
v) Shear with intensity of 0.5 (Shear angle in counter-clockwise direction in degrees)
vi) Zoom within 20% of the original photo randomly
vii) Flip horizontally in a random way
viii) Fill Point outside the boundaries of input with the nearest data

Figure 65: Example photos after augmentation

[29] Definition of augmentation refer to Keras library, https://keras.io/preprocessing/image/

The following is the summary of our database after augmentation,

Table 2: Statistic of the database

Lebel	**Origin**	**Weight**[30]	**1st Augmentation**	**Weight**
BV	138	1.891304348	799	1.836045056
CA	137	1.905109489	808	1.815594059
CB	140	1.864285714	806	1.820099256
CG	182	1.434065934	1055	1.390521327
CJ	261	1	1467	1
FA	209	1.248803828	1194	1.228643216
HC	254	1.027559055	1437	1.020876827
HR	158	1.651898734	912	1.608552632
MC	209	1.248803828	1204	1.218438538
SL	231	1.12987013	1316	1.114741641
Total	1919		10998	

7.1.2. Result

We use the database to retrain the MobileNet V1 and upload the retrained model to the mobile application for taking the real leave for testing in order to understand the performance in the real scenario. However, some plants have already be damaged by the typhoon and replaced, some of them have changed the color because of the seasonal shift, we could not perform testing for all kinds of leaves.

Table 3: Result of recognition with original data

Trial / **Label**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Weight**	**Average percentage**	**Weighted Average Percentage**[31]	**Pass Rate**[32]
BV	76.4	43.5	0	0	29.4	75.4	12.5	69	22.5	47.8	1.891304348	37.65	71.2076087	80%
CB	50	45.6	78.4	68.9	57	78.4	28.6	36.6	58.7	77.7	1.864285714	57.99	108.1099286	100%
CG	77.7	75.8	75.9	78.4	76.3	78.2	77.3	77.8	76	66.3	1.434065934	75.97	108.945989	100%
FA	12.5	0	18.3	0	0	0	0	0	0	47	1.248803828	7.78	9.71569378	20%
HC	0	0	0	15.2	33.2	0	0	0	0	0	1.027559055	4.84	4.973385827	20%
HR	39	32.2	60	76.1	69.5	52.1	73.5	69	67.4	75.5	1.651898734	61.43	101.4761392	100%
SL	25.4	10.1	0	0	21.5	13.2	33	40	19.8	14.5	1.12987013	17.75	20.05519481	80%

[30] $Weight$ is calculated as follows: $\frac{The\ number\ of\ photos\ with\ label\ y}{The\ maxium\ number\ of\ photos\ in\ the\ group}$

[31] $Weighted\ Average\ Percentage$ is calculated as follows: $\frac{Sum\ of\ percentage\ in\ all\ trial}{10} \times Weight$

[32] Pass Rate is calculated as follows: $\frac{The\ number\ of\ time\ lebel\ matches\ the\ top\ 3\ result\ of\ the\ recognition}{10}$, which can be considered the number of chance of non-zero recognition score within 10 trials.

Table 4: Result of recognition with data after 1st augmentation

Trial / Label	1	2	3	4	5	6	7	8	9	10	Weight	Average percentage	Weighted Average Percentage	Pass Rate
BV	77.6	57.7	78	78.6	16.6	35.4	31.6	78.1	64.1	78.6	1.836045056	59.63	109.4833667	100%
CB	46.6	0	26.9	54.1	62.8	10.8	53.1	0	11.3	48.3	1.820099256	31.39	57.13291563	80%
CG	78.7	78.7	78.7	78.7	78.7	78.7	78.7	78.7	78.7	78.7	1.390521327	78.7	109.4340284	100%
FA	0	0	65.2	26.4	78.6	0	34.1	76.3	66.6	11.9	1.228643216	35.91	44.12057789	70%
HC	28.1	11.4	36.3	0	0	0	29.6	0	72.4	44.3	1.020876827	22.21	22.67367432	60%
HR	78.6	78.6	78.1	66.3	78.6	64.3	75.6	78.7	73.1	78.5	1.608552632	75.04	120.7057895	100%
SL	44.7	20.8	22.9	16.2	48.3	0	31.5	27.8	0	69.2	1.114741641	28.14	31.36882979	80%

Table 5: Summary of Weighted Average Percentage in different augmentation

Label	origin	1st	Net Result (1st vs origin)
BV	71.2076087	109.4833667	38.27575801
CB	108.1099286	57.13291563	-50.97701294
CG	108.945989	109.4340284	0.488039425
FA	9.71569378	44.12057789	34.40488411
HC	4.973385827	22.67367432	17.70028849
HR	101.4761392	120.7057895	19.22965023
SL	20.05519481	31.36882979	11.31363498
Average	60.64056	70.70274	10.06218

Table 6: Summary of Average Percentage in different augmentation

Label	Origin	1st	Net Result (1st vs origin)
BV	37.65	59.63	21.98
CB	57.99	31.39	-26.6
CG	75.97	78.7	2.73
FA	7.78	35.91	28.13
HC	4.84	22.21	17.37
HR	61.43	75.04	13.61
SL	17.75	28.14	10.39
Average	37.63	47.28857	9.658571

Table 7: Summary of Pass Rate in different augmentation

Label	origin	1st	Net Result (1st vs origin)
BV	80%	100%	20%
CB	100%	80%	-20%
CG	100%	100%	0%
FA	20%	70%	50%
HC	20%	60%	40%
HR	100%	100%	0%
SL	80%	80%	0%
Average PR	71%	84%	13%

Weighted Average Percentage (a.k.a. WAP) is the reference average to indicate the recognition performance of a kind of leave when we use to similar or sufficient amount of sample of leave to do the training. Since data collection is the culprit of causing the imbalance amount of training data, we would like to show the weighted average as an indicator to the theoretical performance for the recognition.

In Table 5, all labels have the significant enhancement on the successful recognition in average, except label CB has the drop after the 1st augmentation. Therefore, augmentation like some basic translation, rotation and zoom can impact the recognition and improve the result for most of the image when we have limited resource of training data.

Average Percentage (a.k.a. AP) and Pass Rate (a.k.a. PR) is the indicative of the performance of recognition in the real use case. The best scenario of PR is 100% in all trial, which means user can receive the recognition result of the photo taken within top 3 labelling and we can deliver the correct plant information to user in a higher chance. In Table 7, more than 5 kinds of leave achieve 50% of PR in total, and the average PR is 71% with the origin dataset. After 1st augmentation, the average PR reaches 84% for all 8 kinds of leaves, means users are most likely to receive the plant information they want to know with our application.

7.1.3. Conclusion

Data augmentation provides assistance to retrain the model with the limited resource as it provides more details and features of the training data by simulating the data in different environments.

7.2. Experiment II: Augmentation regarding different lighting condition

In this experiment, we would like to focus on choosing the more appropriate lighting variables in order to improve the method of lighting augmentation and acquire a better performance of recognition.

7.2.1. Specification

We add the different lighting conditions to simulate photo taken in various environment, such as under sun light or in the cloudy day. Every photo, underwent the 1st augmentation, will go through this augmentation with the below formula for 4 times, with different variables of contrast and brightness.

$$[\boldsymbol{R}', \boldsymbol{G}', \boldsymbol{B}'] = c * cnum * [\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}] + [\boldsymbol{\beta}, \boldsymbol{\beta}, \boldsymbol{\beta}]$$

where,

$i)$ $[\boldsymbol{R}', \boldsymbol{G}', \boldsymbol{B}']$ *is the output image*

$ii)$ $c = 0.1$

$iii)$ *cnum is the contrast variable*

$iv)$ $\boldsymbol{\beta}$ *is the brightness variable, which has the same dimension of a color channel in the input image*

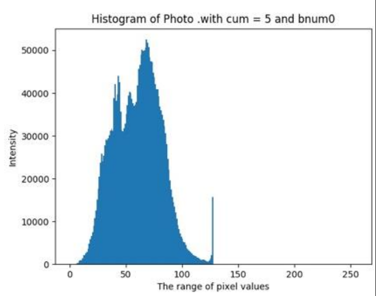
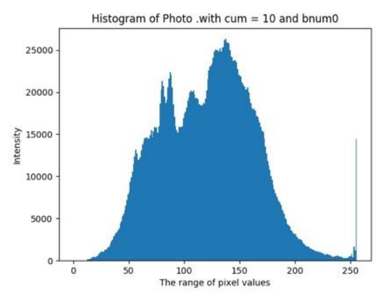
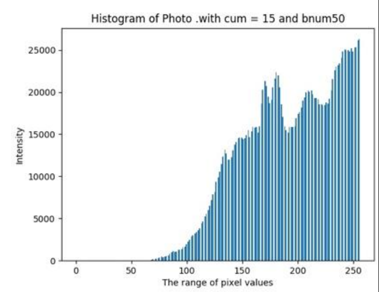
$v)$ $[\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B}]$ *is the input image*

We choose two sets of variables of the lighting condition. The first set of variables will have a more extreme effect on lighting. The second set of variables are chosen based on the histogram of the photos after lighting-augmented, such that *the convex shape will be within 25% measured from the center of the plot*. After that, we will augment the images and generate two set of image data and retrained two models. The experiment will be carried out to compare the score of recognition in two models by labelling the same set of photos which has never been used in training of the model.

7.2.1.1. Image Histogram

Image histogram is commonly used in photography to understand if a photo is overexposed, underexposed or correctly exposed. The histogram will first convert the rgb chanel to grayscale pixel in range of [0, 256] and plot the intensity of each grayscale pixel. After that, we will see a convex shape of the main object in the image. To determine the exposure, we can check if the convex shape is close to the center of the plot. If it is shifted in either right or left, then the photo may be overexposed or underexposed.

Table 8: Example of histograms

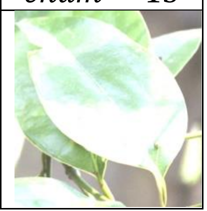
Under exposed	Correctly exposed	Overexposed
		
The convex shape is shifted to the left from the center of the plot.	The convex shape is laid at the center of the plot.	The convex shape is shifted to the right from the center of the plot.

7.2.1.2. Lighting Variables

After considering the histogram of the images, we come up with two set of variables. Two sets of image data will be generated with the selected variables. Demonstration of the images is as follows,

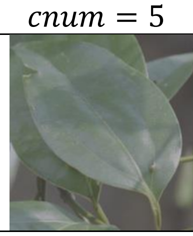
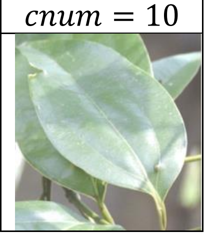
SET 1: Extreme lighting condition

Table 9: illustration of images of set 1

Iteration	0 (Origin)	1	2	3	4
Variables	$c = 1$ $\beta = 0$ $cnum = 1$	$c = 0.1$ $\beta = 0$ $cnum = 5$	$c = 0.1$ $\beta = 50$ $cnum = 5$	$c = 0.1$ $\beta = 0$ $cnum = 15$	$c = 0.1$ $\beta = 50$ $cnum = 15$
Result image					

SET 2: Acceptable range of exposure measured with histogram

Table 10: illustration of images of set 2

Iteration	0 (Origin)	1	2	3	4
Variables	$c = 1$ $\beta = 0$ $cnum = 1$	$c = 0.1$ $\beta = 50$ $cnum = 5$	$c = 0.1$ $\beta = 10$ $cnum = 10$	$c = 0.1$ $\beta = 30$ $cnum = 10$	$c = 0.1$ $\beta = 50$ $cnum = 10$
Result image					

7.2.2. Result

Detailed result is enclosed in the appendix 13.4.

Table 11: Score of prediction in different models

Column:	1	2		3		
File name	Score from model retrained with 1st augmented data	Score from model retrained with augmentation with set 1 variable	Net Result (Col: 2-1)	Score from model retrained with augmentation with set 2 variable	Net Result (Col: 3-1)	Net Result (Col: 3-2)
bv_bright.jpg	0.99952	0.99925	-0.00027	0.99987	0.00035	0.00062
bv_dark.jpg	0.99715	0.99834	0.00119	0.99957	0.00242	0.00123
ca_bright.jpg	0.24871	0.5896	0.34089	0.5528	-0.16704	-0.50793
ca_dark.jpg	0.3984	0.59977	0.20137	0.43227	-0.08022	-0.28159
cb_bright.jpg	0.95536	0.88185	-0.07351	0.93211	-0.02325	0.05026
cb_dark.jpg	0.99997	0.99972	-0.00025	0.99996	-1E-05	0.00024
cg_bright.jpg	0.98583	0.99535	0.00952	0.99438	0.00855	-0.00097
cg_dark.jpg	0.99975	0.99981	6E-05	0.99973	-2E-05	-8E-05
cj_bright.jpg	0.9998	0.99997	0.00017	0.99982	2E-05	-0.00015
cj_dark.jpg	0.99809	0.99906	0.00097	0.99671	-0.00138	-0.00235
fa_bright.jpg	0.97914	0.98521	0.00607	0.97855	-0.00059	-0.00666
fa_dark.jpg	0.86202	0.87345	0.01143	0.54897	-0.31305	-0.32448
hc_bright.jpg	0.99972	0.9958	-0.00392	0.99956	-0.00016	0.00376
hc_dark.jpg	0.99984	0.99989	5E-05	0.99953	-0.00031	-0.00036
hr_bright.jpg	0.9924	0.91952	-0.07288	0.88863	-0.10377	-0.03089
hr_dark.jpg	0.99737	0.99214	-0.00523	0.99895	0.00158	0.00681
mc_bright.jpg	0.94465	0.97052	0.02587	0.95077	0.00612	-0.01975
mc_dark.jpg	0.48928	0.32452	-0.16476	0.71835	0.22907	0.39383
sl_bright.jpg	0.87673	0.95879	0.08206	0.50232	-0.37441	-0.45647
sl_dark.jpg	0.86204	0.93595	0.07391	0.48486	-0.37718	-0.45109
AVERAGE	0.8792885	0.9009255	0.021637	0.85483	-0.0304	-0.0460955

According to the result shown in Table 11, the model retrained with origin dataset (image processed with 1st augmentation), the average prediction accuracy is 87.9%. Using the model retrained with data using SET 1 variables has 2% of improvement in average from the origin dataset, and the average score is 90%. However, the model retrained with image using SET 2 variables turns out a drop of score for 4% from original score, and its average score is 85%.

Hence, lighting variables from SET 1 is a better option for augmentation regarding the lighting.

7.2.3. Discussion

Originally, we believed that choosing the lighting variables with the help of histogram will be result in a better improvement of data augmentation, since we can prevent the circumstance of being overexposed or underexposed in the image, and the augmented image would make more sense to human perception. However, the experimental result turns out that a more aggressive choice of lighting variables may have the positive impact whereas the more rational choice can deteriorate the performance of recognition. One of the possible inferences can be the image data augmented with lighting variables from SET 2 may not cause huge effect to the photo, in which the difference in the image may not sufficient enough to trigger the model to make a better classification for the leaves. On the other hand, the image data augmented with lighting variables from SET 1 have adequate influence and provide more features or details to the model, so that the result of recognition is improved. For instance, we can observe that in Table 9, the image in 4th iteration has shown the situation of being overexposed, but the shape of the leaf could be seen more obvious than other images in the same table. Therefore, a certain clue has given to the model for making guess on the leaves.

All in all, we can apply a more extreme and aggressive variation regarding lighting condition in data augmentation.

7.3. Experiment III: Does color matter in the recognition?

What if the color of leaves changes along the season? Does our application or the model be able to deal with the tricky situation and yield the correct result of the recognition? In reality, the leaves do not always remain their color since different weather conditions will affect the growth of the plants and change the color them. Hence, we would like to figure out how the change will interfere the recognition.

7.3.1. Specification of color change

We adopt functions in OpenCV library to change the color of input images in order to simulate different color changes in reality. We will try on a gray image and swap the RGB channel in the image to make the color change. 10 images from all label are chosen to carry out the experiment, see appendix 14.4.1 for details.

In OpenCV, the image will be presented in the order of [B, G, R] instead of [R, G, B]. As a result, cv2.merge((b,g,r)) will yield the same image as original image, and cv2.merge((r,g,b)) is the different image instead.

Table 12: Illustration of color change of label BV

Origin Image	***Function / Method***	***Image with color change***
	cv2.cvtColor(photo, cv2.COLOR_BGR2GRAY)	
	cv2.merge((r,g,b))	
	cv2.merge((r,b,g))	
	cv2.merge((b,r,g))	
	cv2.merge((b,g,r))	
	cv2.merge((g,r,b))	
	cv2.merge((g,b,r))	

7.3.2. Result

Predictions' score from three retrained model is included in the appendix 14.4

Figure 66: Statistic showing the average improvement of prediction in model retrained with original data

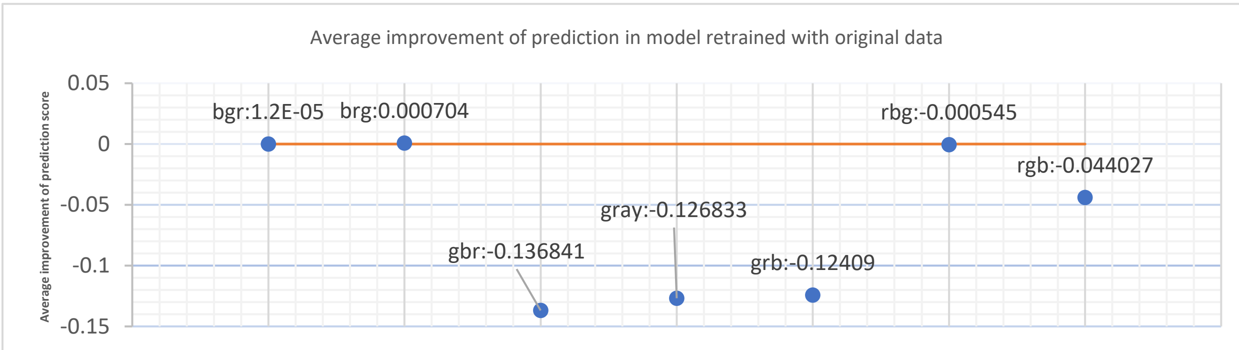


Figure 67: Statistic showing the average improvement of prediction in model retrained with data after 1st augmentation

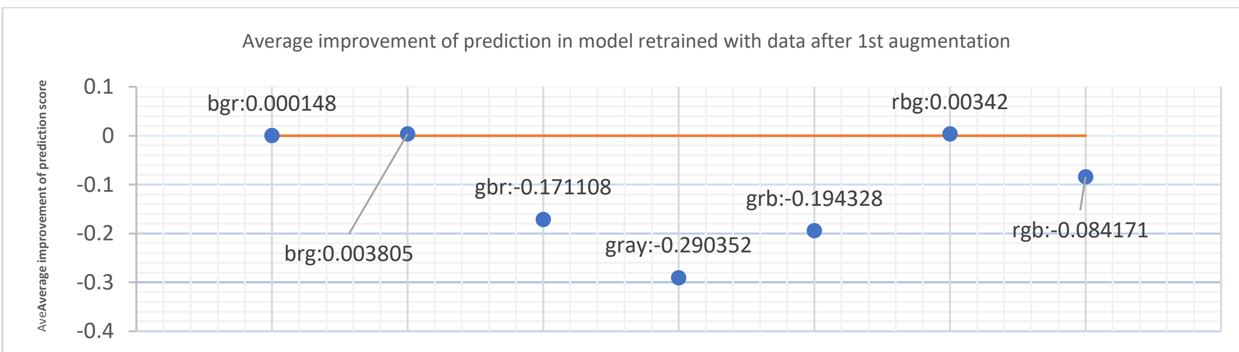
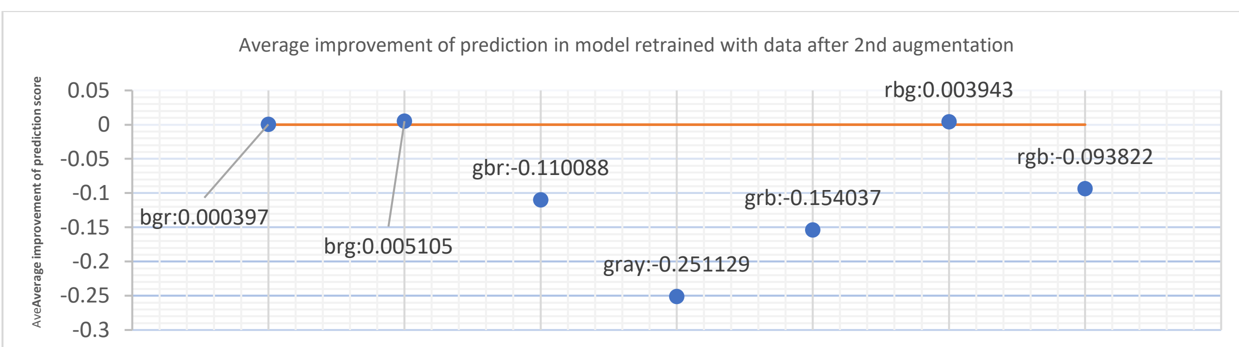


Figure 68: Statistic showing average improvement of prediction in model retrained with data after 2nd augmentation



The above figures show the improvement of prediction score of color-changed images with model retrained by different datasets.

The idea of calculation is that,

(Step 1) we first figure out the difference of prediction score of each photo with color-changed against the prediction score of the origin photo.

(Step 2) we take the average of all the difference of the prediction score with respect to the methods of changing color.

For method **gbr**, **gray, grb** and **rgb**, they all have the negative effect in the prediction score in different models, range from -0.044027 (In Figure 66, method rbg) to -0.290352 (In Figure 67, method gray). Meanwhile, **brg** and **rbg** have the improvement in the prediction score in all retrained models, except a minor score deduction has been reflected of method **rbg** with respect to the model retrained with origin data. The maximum increase is 0.0051 (In Figure 68, method brg). As for the method **bgr**, a tiny improvement has been recorded with the maximum increase of 0.000397 away from the origin prediction score.

Method **gray** had the most significant decrease among all other methods, from -0.126833 (in Figure 66) to -0.290352 (in Figure 67).

All in all, except method **bgr**, 4 methods have the negative effect and 2 methods have the positive influence in the experiments.

7.3.3. Discussion

By the experimental result, the change of color in image does affect the prediction result. Minor improvement of score is found, whereas major deduction of score is recorded. The reason of worse prediction is that color conversion leads to the loss of information and knowledge in the image, which causes the interruption in classification. However, an interesting founding is that some color-changed images still have the improvement in the prediction score, like **rbg**. If we check the resultant images of **rbg**, a common change of color of leaves along different seasons (especially when it is stepping into autumn) can be observed. Since the change is close to the reality, we might have the prediction that our models can handle the real situation, when the color of leaves changes.

But for the gray image, the model will be result in the incorrect guess and cause the relatively huge decrease in the experiment. Some researches show that different algorithms of color conversion from color image to gray image have different impact. For example, method "Gleam" of color conversion will have the best result with respect to facial and object recognition; Method Luminance is preferable for texture recognition. [3] In OpenCV, the conversion is based on the formula[33] that have not been tested in the research, and our result shows that there might be a negative impact range from -0.12 to -0.29 away from the origin prediction score. The drop should be caused by the same reason of information loss during the color conversion and the model is difficult to figure out the features about the leave. Although the result shows that gray image might have the negative result on the improvement of recognition, the question for if there exists a conversion method of color-gray image that will be suitable for recognizing feature of leaves is still open in the future.

On the other hand, the reason of the prediction still works on the color-changed leaves photo can be inferred that the features like the contour or the shape are still able to be discovered by the neural network. Color might be one of the features that the model will

[33] The definition of formula is stated as follows, RGB[A] to Gray: $Y \leftarrow 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$, See reference [4] for details.

apply during learning and recognition. However, mere the change of color is not enough to twist the prediction. We will make an attempt to discuss the features that the model will use for learning, training and predicting in the later part of visualization.

In conclusion, although the maximum score deduction of -0.29 is found in our experiment, the model is believed to have the rest of 70% of chance to hit the prediction of correct label, and our application should be able to handle the tricky case in reality.

7.4. Experiment IV: Can the dried leaves be recognized?

The appearance of leaves changes along the time, even it will be damaged if it detaches from the plants. This part of experiment will find out the possibility of recognizing the dried leaves. Nevertheless, limitation still existed in this experiment, such that collecting the dried leaves is not an easy task because the appearance of some dried leaves may be totally be different from the origin leaves, therefore the complexity of data collection is increased. The result here cannot be generalized to all kinds of leaves.

7.4.1. Specification of dried leaves

We choose leaves label BV since it is believed BV has a special shape among other leaves in our database.

Table 13: Illustration of dried images of BV

File Name	bv00011_dried_1	bv00011_dried_2	bv00011_dried_3	bv00011_dried_4
Image				

7.4.2. Result

Table 14: Prediction of dried leaves with respect to different models

File name	Top 1 label	Top 1 score	Top 2 label	Top 2 score	Top 3 label	Top 3 score	Top 4 label	Top 4 score	Top 5 label	Top 5 score
Predict with model retrained with origin dataset										
bv00011_dried_1.jpg	bv	0.99253	hc	0.00488	cg	0.00234	cj	0.00023	hr	0.00002
bv00011_dried_2.jpg	hr	0.45673	bv	0.39715	hc	0.12549	cj	0.0142	cg	0.00639
bv00011_dried_3.jpg	hc	0.91146	bv	0.08575	cg	0.00164	cj	0.00106	cb	0.00009
bv00011_dried_4.jpg	cg	0.48314	bv	0.18468	hc	0.14797	cb	0.09793	sl	0.04912
Predict with model retrained with 1st augmented data										
bv00011_dried_1.jpg	bv	0.99859	cj	0.0008	hc	0.00041	hr	0.00013	cg	0.00006
bv00011_dried_2.jpg	hr	0.88454	hc	0.07329	bv	0.02618	cj	0.01586	cg	0.00013
bv00011_dried_3.jpg	hc	0.9984	cj	0.0008	bv	0.00068	cg	0.0001	cb	0.00001
bv00011_dried_4.jpg	bv	0.90092	cb	0.07039	sl	0.01974	hc	0.00379	hr	0.00316
Predict with model retrained with 2nd augmented data										
bv00011_dried_1.jpg	bv	0.99993	cj	0.00005	fa	0.00001	hc	0.00001	cg	0.00001
bv00011_dried_2.jpg	bv	0.99984	hr	0.00015	cj	0.00001	fa	0	hc	0
bv00011_dried_3.jpg	bv	0.8279	hc	0.13363	cj	0.02803	cg	0.01014	fa	0.00025
bv00011_dried_4.jpg	bv	0.97029	hc	0.00771	cg	0.00762	cb	0.00672	hr	0.00365

The above result shows that the model retrained with 2nd augmented data have the best performance of recognition as 100% correct guess. The rest of the performance have merely 50% of correct guess.

7.4.3. Discussion

The inferred explanation of 100% correct guess could be the light condition revealing more details related to the shape of the leaves, so that the dried leaves could still be able to be recognized, no matter the appearance.

If we adopt the model retrained with origin data and model retrained with 1st augmented data, we might have 50% of correct guessing. However, it is believed that the sample of testing is not sufficiently large enough to justify the claim of 50/50. Further experiment and data collection have to be done to prove the situation.

Furthermore, we will try to demonstrate the inferred explanation in the later part of visualization as well.

7.5. Investigation: Model Visualization

Deep learning has been applied in many different areas for example trend prediction, image recognition or the text classification because of its well-known succeed. However, explaining the reason of how the deep learning models work is still a challenging topic. One of the means to interpret the rationale behind the deep learning is by visualizing the abstract features that have been extracted by the model in order to perform the classification or recognition on different object. The technique of visualization of the trained model have been supported and adopted by some researches on justification since it is human-understandable, and we can try to understand the myth in the black box. [5]

Moreover, as we are utilizing the technique of transfer learning, even though different models have been "retrained" in the above experiments and different recognition results have been presented, there is no variation of the learned variables (weights in kernel) in all the layers before the classifying layer (SoftMax). It means that the layer of performing classification will take the same set of learned features as the knowledge of a particular label. Therefore, we can consider the layers before the classifier grouping as a machine, which takes an image as input and outputs the features inside the image. The concept is shown in the image below,

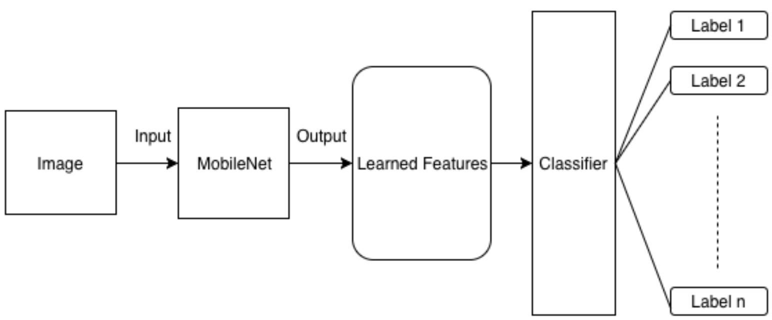


Figure 69: Concept of considering the layers grouping as a machine

Therefore, we can observe and investigate

i) what features does MobileNet learn and,

ii) how the variation of input data may possibly affect the learned features.

7.5.1. Overview of layers in MobileNet

We have mentioned the elementary component “depthwise separable convolution” of MobileNet in the previous chapter. Now, we are going to scrutinize the computation of the first layer in MobileNet, pointing out the exact location of visualization.

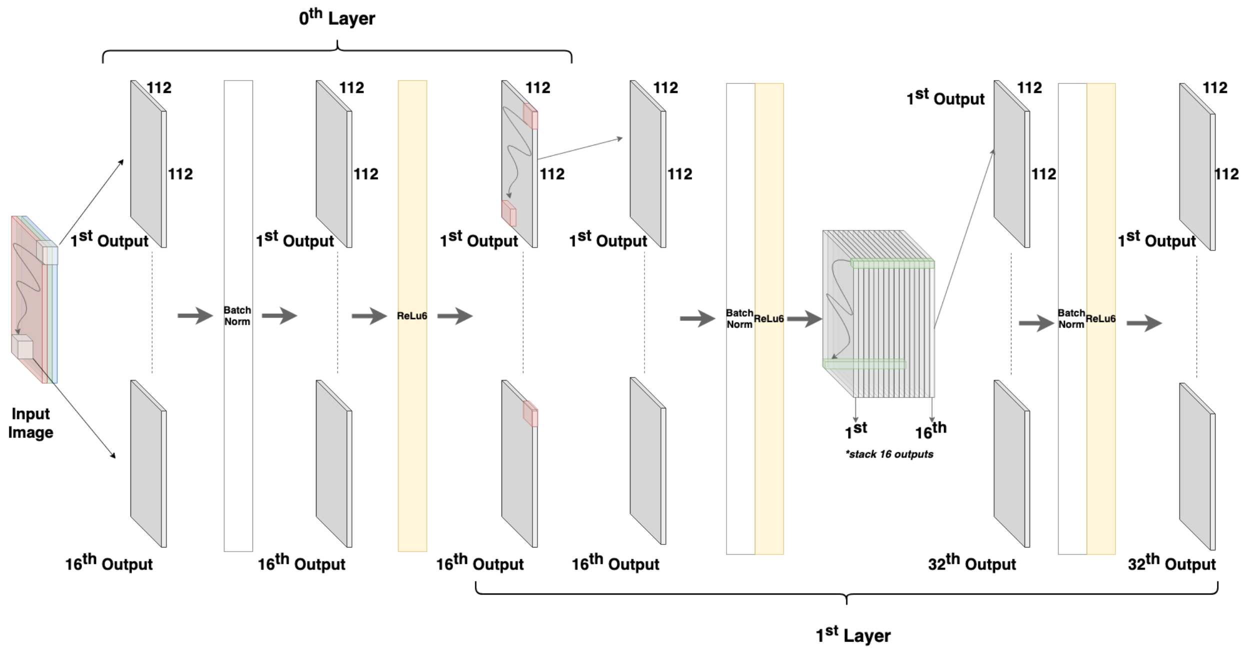


Figure 70: Illustration of the 0-th and 1-st layer of MobileNet

When the image (with size of 224*224) is input to MobileNet for classification, it first passes through the 0-th layer and undergoes a complete computation of convolution with the stride of 2. The gray cube (its size is 3*3*3) inside the input image in Figure 70 is the kernel of convolution.16 different of kernels will walk around the same input image. After the computation of convolution, 16 outputs will be yielded and a batch normalization (denoted batch norm in the figure) will be implemented to the outputs and followed by the RELU6 activation function. The objective of batch normalization is to optimize the training process, since internal covariate shift will occur without normalization.[6] Therefore, it might not be useful in the process of classification. As for the RELU6[34], it is the modified activation function of RELU (**<u>Re</u>**ctified **<u>L</u>**inear **<u>U</u>**nits) of taking the minimal value in

[34] Re:LU 6 is definded as y = min(max(x, 0), 6).

between the maximum output of RELU and 6.[7] Finally, 16 outputs of grey image (its size is 112*112) will be generated for further computation in the later stage.

In the 1-st layer, 16 grey images will be taken as input for the computation of depthwise convolution with stride of 1 in the beginning. The red square (with size of 3*3) in the 1st output in Figure 70 indicates the kernel of depthwise convolution. Another 16 outputs will be produced and undergo the batch normalization and RELU6 activation, stacking up as a cube , a matrix in 3 dimensions (its shape is 112*112*3) for pointwise convolution. The green cuboid (with size of 1*1*32) in Figure 70 represents the kernel of pointwise convolution. Computing with the stride of 1, the convolution will yield 32 outputs as 32 different kernels will be applied in the operation, and the batch normalization and RELU6 activation will be performed afterward as well.

The whole MobileNet will repeat the above depthwise separable convolution and produce a long vector of size of 1*1*1024 for the SoftMax classifier. A remark for the above description is that we use 0.5 size of the original model so that the number of outputs in every layer will be diminished by half from the specification mentioned in the origin paper.

The investigation of visualization in this section will focus on outputs after RELU6 activation, which its position has been highlighted in yellow in Figure 70. Only the first layer, including the 0-th, will be shown in this section since the outputs in the later stage of model will finally be encoded to some matrices which are difficult for human to recognize.

7.5.2. Visualization of learned features

In this section, we will reveal the black box of MobileNet and make an attempt to summarize what features are extracted during the classification. We will take the bright leaves photos mentioned in appendix 14.3.3 as the demonstration of showing the captured features.

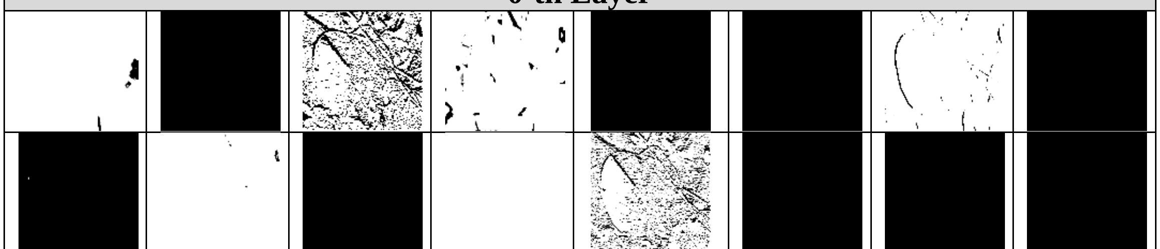
Table 15: Features in the first layer of leaf label ***BV***

Input Image

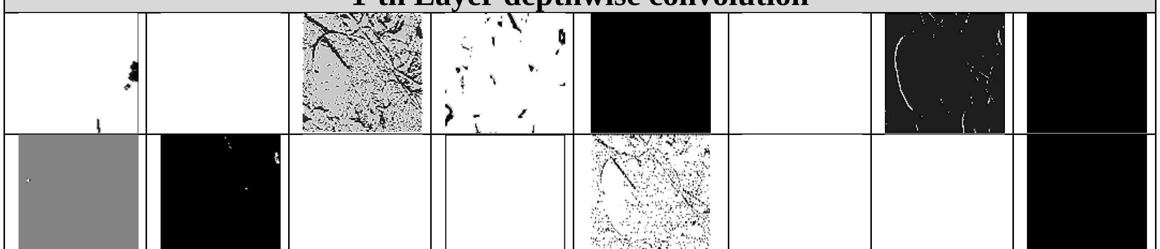
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution

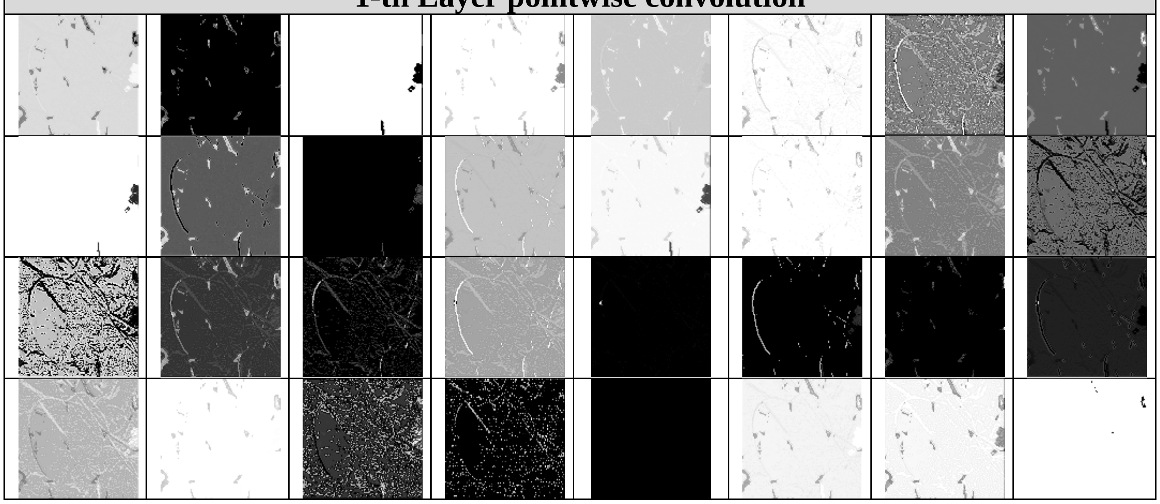


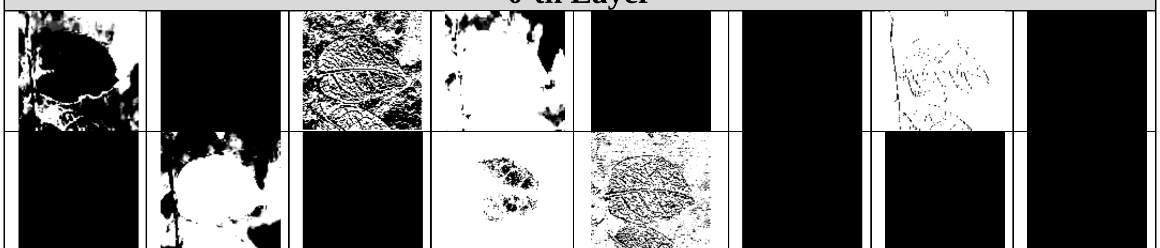
Table 16: Features in the first layer of leaf label ***CG***

Input Image

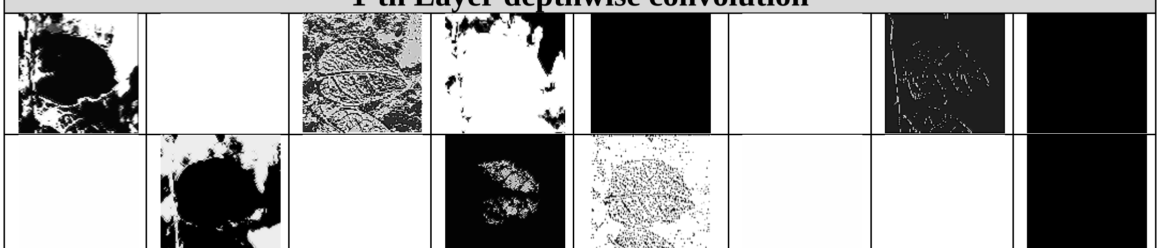
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution

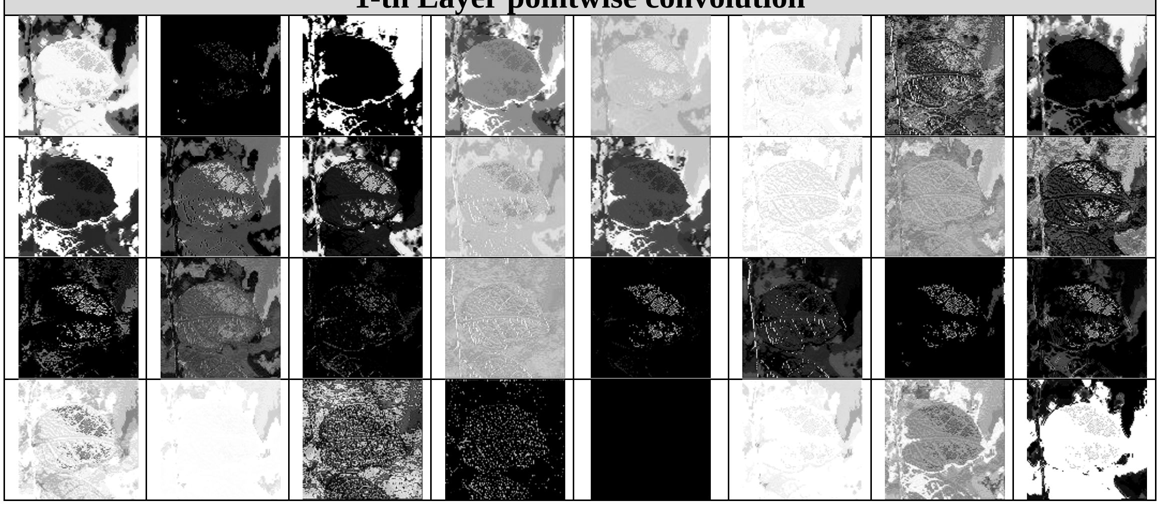


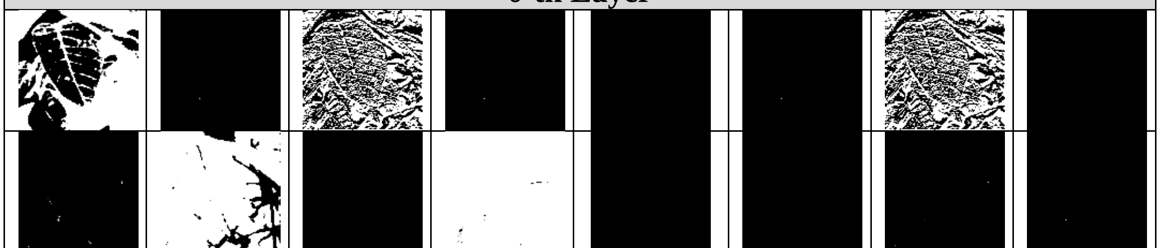
Table 17: Features in the first layer of leaf label ***CA***

Input Image

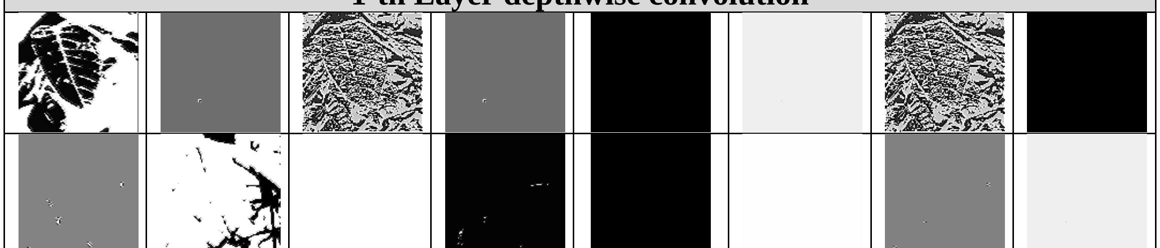
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

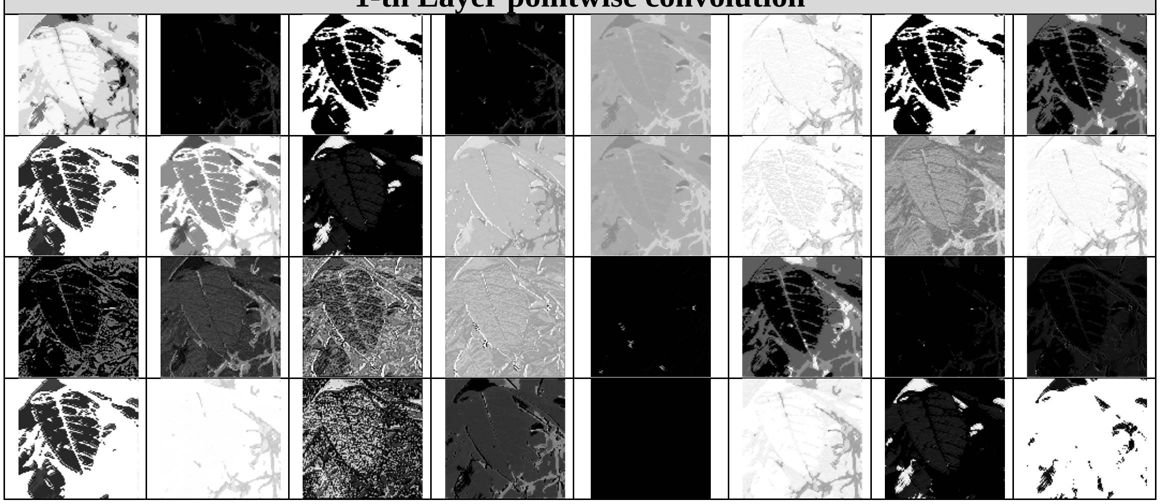
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



In the above tables, the whole process of image passing though the 0-th and 1-st layer of MobileNet is shown respectively. The visualization of label BV, CG and CA will be shown in above.

First of all, we can observe that a number of outputs which are totally in black or in white are presented in every layer. According to the definition of RELU6 activation function, it will constrain every data entry (pixel) in the value between 0.0 and 6.0. If the value of data entry is larger than 6.0, the pixel will be mapped to 6.0 as the result of activation. Therefore, the black image may mean that there exist features that the particular channel is trying to extract in the whole input image. However, it is still the secret that we might not know what feature is related to it, according to the result of activation. As for the white images, it is obvious that there is no activation of the particular channel for some unknown features. As we are using the pretrained model, which is trained with ImageNet Dataset, some of the features or channels are not suitable for recognizing the image of leaves. Hence, a number of black and white image exists.

Apart from the black images, we can still try to infer the learned features from the result of activation in the output channel. In the index 1 of the above tables, it is believed that the model is extracting the feature related to dark *color* in the images, as only the area of leaf has been activated in the output images. The area of leaf’s stem (see the input image in Table 17 for the yellow stem), has no activation at all. Hence, the channel is scanning the pattern of the object in input image regarding to the color. As for the same channel in Table 15, the output image, however, is almost in white. The reason of it can be inferred as the color of the leaf is not dark enough, when it compares to the input image in Table 16 and Table 17. Therefore, no shape of leaf is extracted in terms of colors.

Moreover, *edges* in the image are also the crucial features that the model will extract for recognition. As we can see, for instance in the channel of index 3 and 7 in the 0-th layer, different degree of edges has been extracted. Even in the later layer, we can observe the edges in different greyscale outputs. With the help of the edges, the classifier could be able to learn the *shape*, *texture and* *contour* for different leaves as well. Edge detection is the traditional method in computer vision for image analysis, in order to reduce the amount of data but still remain the important information for the targeted object. [8] In the era of neural network, the similar approach can also be embodied in the learned features.

With the means of visualization, we are able to analyze the result of the experiments we carried before and make an attempt to provide answers for the following three questions: How will (i) different lighting condition, (ii) color-changed leaves and (iii) dried leaves affect the recognition?

7.5.3. Visualization regarding different lighting conditions

In chapter 7.2, we conducted an experiment to compare the result of recognition of two-sets lighting variables, changing the image with respect to its brightness and contrast in data augmentation. The result shows that the model retrained with the image data augmented with a wide range of variables in the formula mentioned in 7.2.1 have a better score (+0.021637 against the original score) than the range chosen in an rational way according to the histogram (-0.0460955 against the original score). As a result, we can explore the difference of the channel outputs between the images augmented with various lighting variables through visualization technique in this section. We will demonstrate the following four pairs cunm and bunm, which are selected from SET 1 and SET 2 stated in the specification in the above experiment.

Table 18: Features in the first layer of leaf label CB. Input image is augmented with cnum = 5 and bnum = 0

Input Image

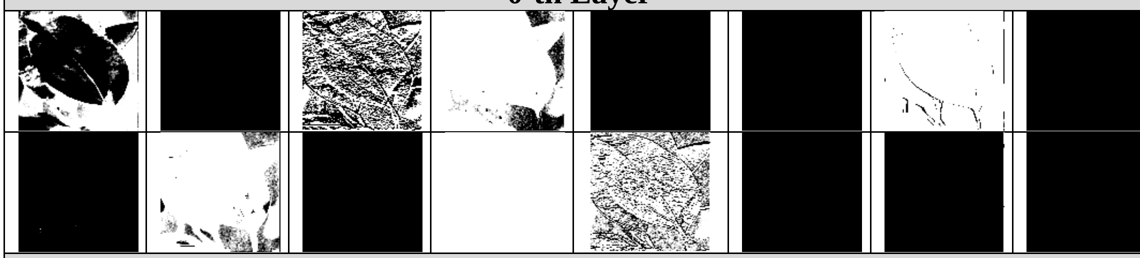
Index of output in a layer:

0-th Layer							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

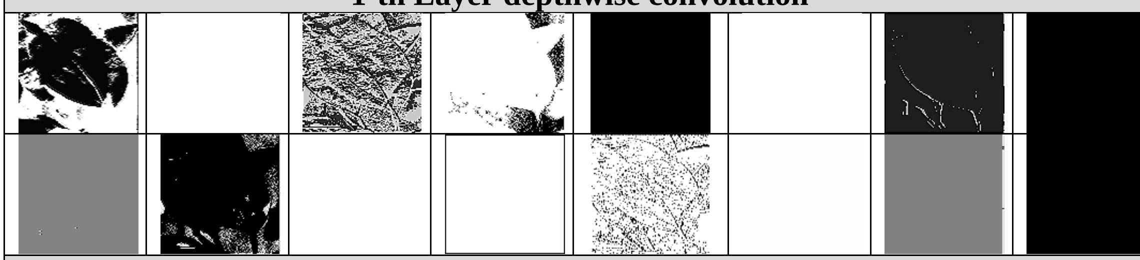
1-th Layer Depthwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

1-th Layer pointwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution

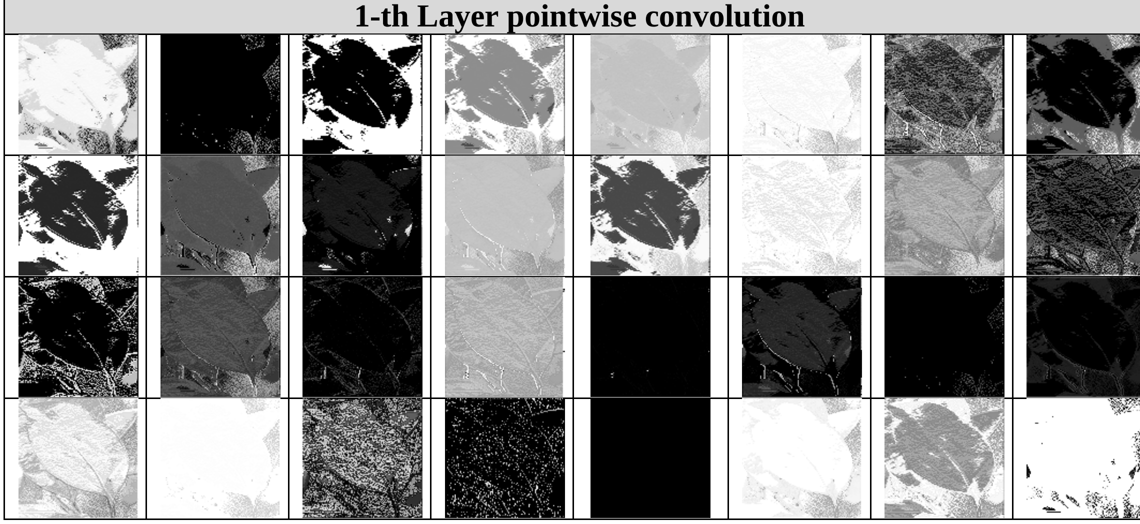


Table 19: Features in the first layer of leaf label CB. Input image is augmented with cnum = 15 and bnum = 50

Input Image

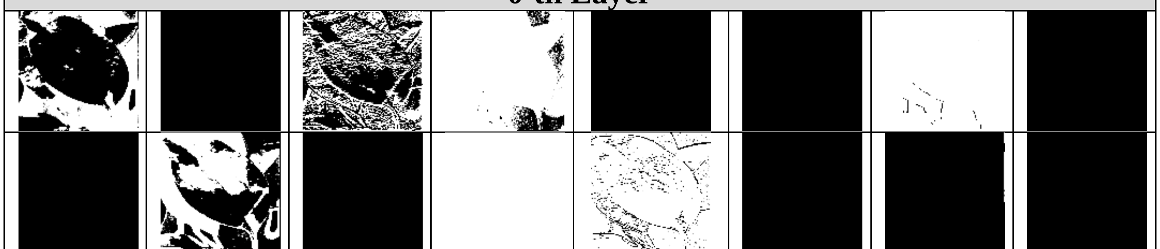
Index of output in a layer:

0-th Layer							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

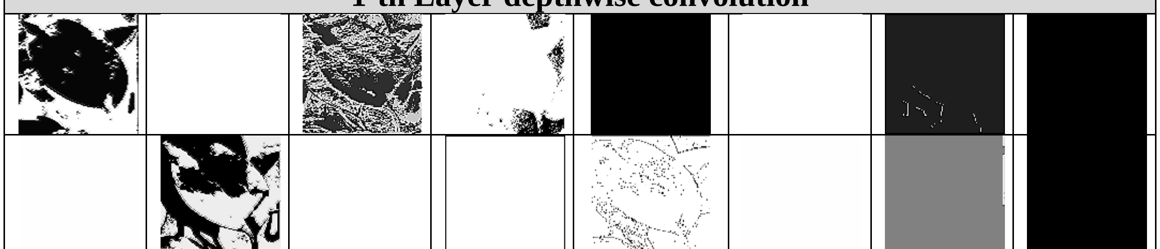
1-th Layer Depthwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

1-th Layer pointwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution

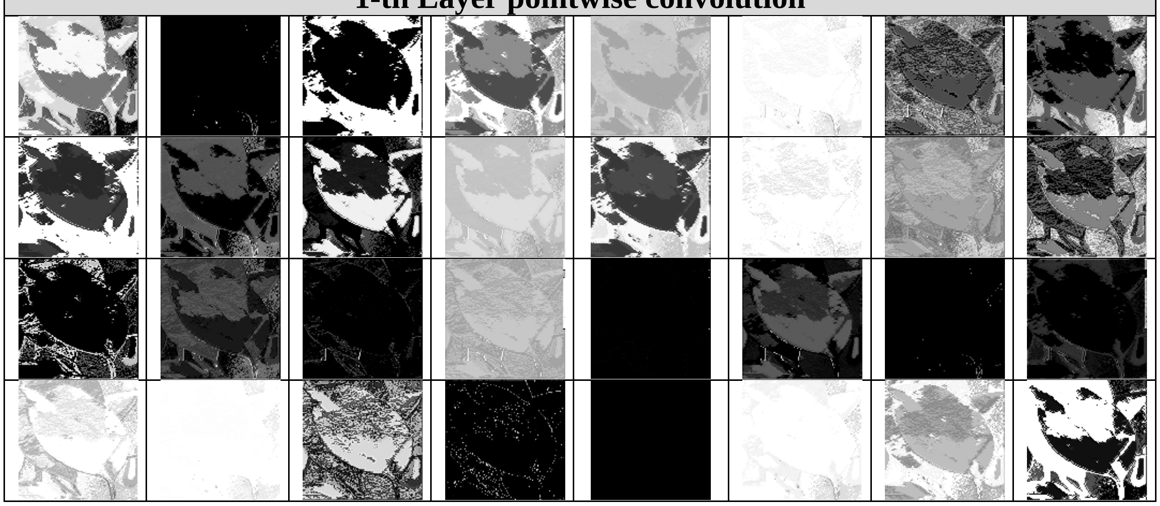


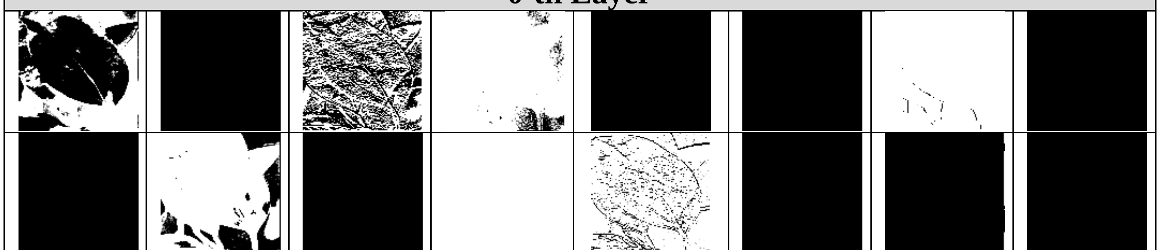
Table 20: Features in the first layer of leaf label CB. Input image is augmented with cnum = 10 and bnum = 50

Input Image

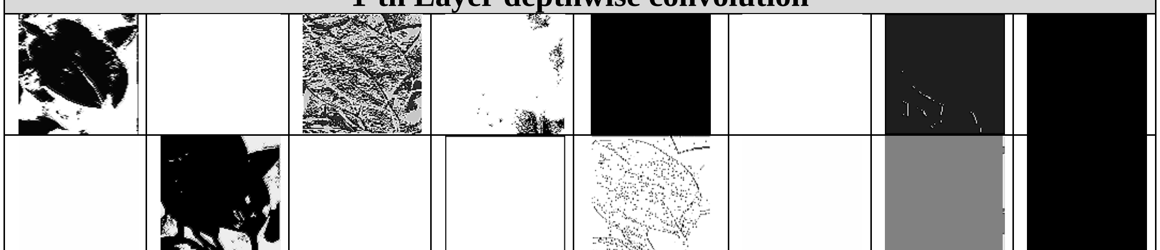
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution

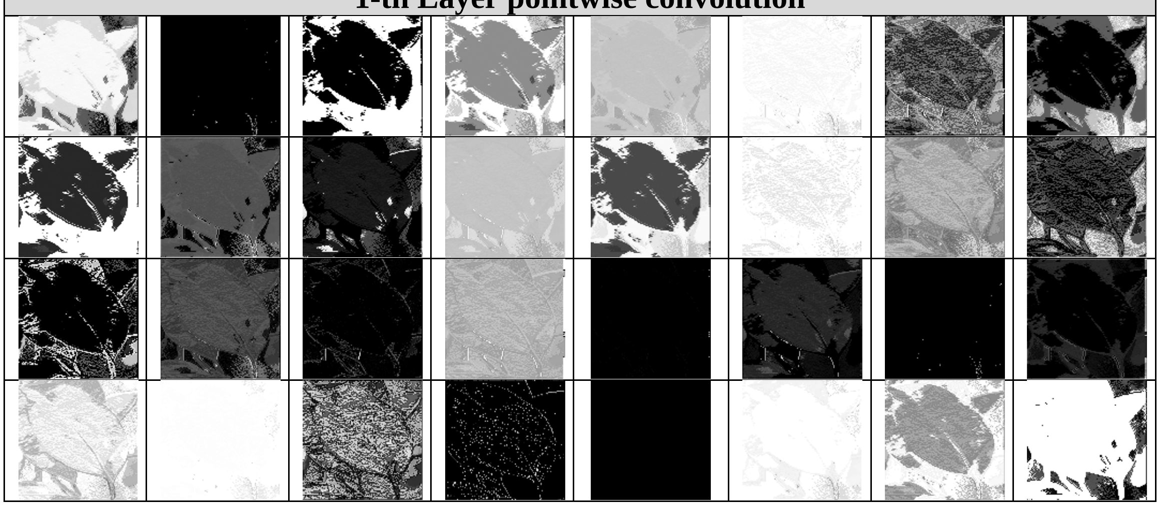


Table 21: Features in the first layer of leaf label CB. Input image is augmented with cnum = 5 and bnum = 50

Input Image

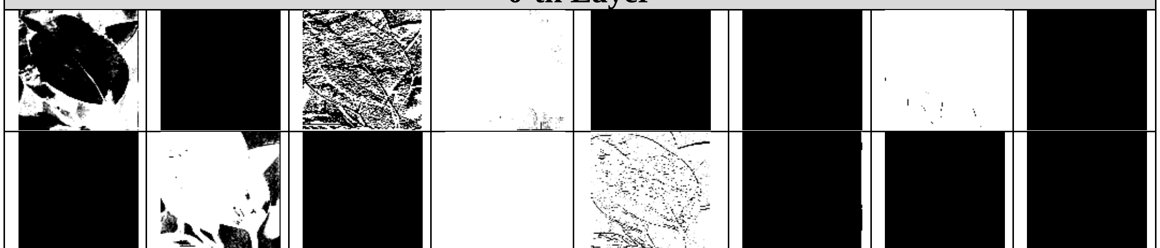
Index of output in a layer:

0-th Layer							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

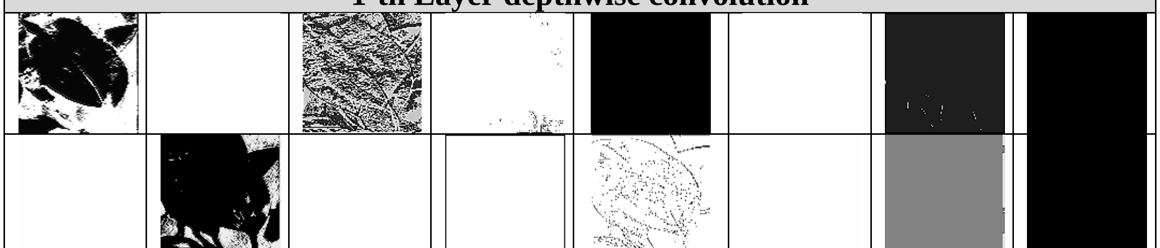
1-th Layer Depthwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

1-th Layer pointwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

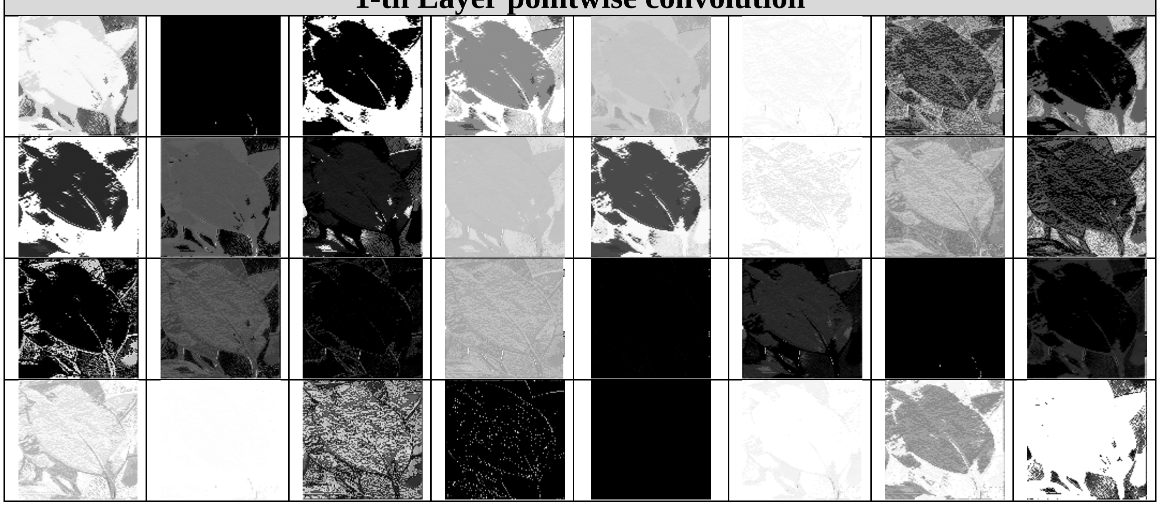
0-th Layer



1-th Layer depthwise convolution

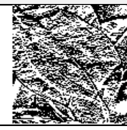
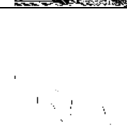


1-th Layer pointwise convolution



First two tables, Table 18 and Table 19, describes the features in the first layer with the input image augmented by the variable from SET 1.Table 20 and Table 21, illustrates the features with the input image augmented using the variables from SET 2.

Table 22: Summary of output channel with index 3,7,10,13 in 0-th layer

Index	SET 1 Tables		SET 2 Tables	
	Table 18	Table 19	Table 20	Table 21
3				
7				
10				
13				

First of all, in the 0-th layer, we can take a look into the output of index 3, 7, 10 and 13[35]. An apparent difference of outputs can be found in the SET 1 tables, whereas a slightly difference occurs in the SET 2 tables. It is believed that the degree of difference in the output channels should mean the degree of additional information or knowledge providing to the model for classification. Hence, more information of the same kind of leaf can be provided if we choose the wide range of lighting variables, as the results shown in SET 1 tables, when it compares to SET 2 tables. The same circumstances can be reflected in the same indices of output channels in the 1-th layer depthwise convolution as well.

Moreover, in the 1-th layer pointwise convolution, the outputs in SET 1 tables will have a more textured grayscale image against the output channels in SET 2 tables. For instance, the output channels of index 4,5,10,11,19,22 and 27[36] in Table 19, that the input image is applied with a more extreme lighting variables in augmentation, are sharper. The shape of the entire leaf can still be observed whereas the extreme lighting condition. However, the

[35] The index table is highlighted in green.
[36] The index table is highlighted in red.

outputs in SET 2 tables are not sharp, leading to the problem of insufficient information for classification.

All in all, the above revealing of the black box supports the reason for considering the wider range of lighting variables. The reason of higher accuracy in retrained models with SET 1 lighting variables can be countenanced by the result of visualization.

7.5.4. Visualization regarding color-changed leaves

In the previous chapter, we carry out an experiment to see whether the model can handle the tricky case of color-changed leaves. In summary, 4 methods of changing color (**gbr**, **gray**, **grb** and **rgb**) have score deduction whereas improvement have been shown in 2 methods of changing color (**brg** and **rbg**). Therefore, we will try to divulge the secret of the model and observe what is actually happening in the first layer by visualization. Our methods of changing color were swapping the **rgb** channel of the original image, the direct effect can be seen after the convolution operation in the 0-th layer. We will demonstrate the output channels, see appendix 14.5 for detailed outputs.

In the demonstration, we will first show the visualization of **bgr** as a reference of origin photo before color-changed. After that **gbr**, **gray**, **grb** and **rgb** will be shown as the comparison and followed by **brg** and **rbg**.

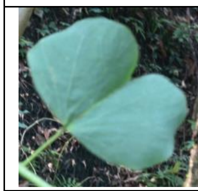
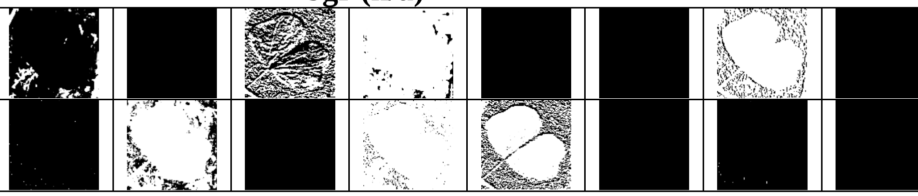
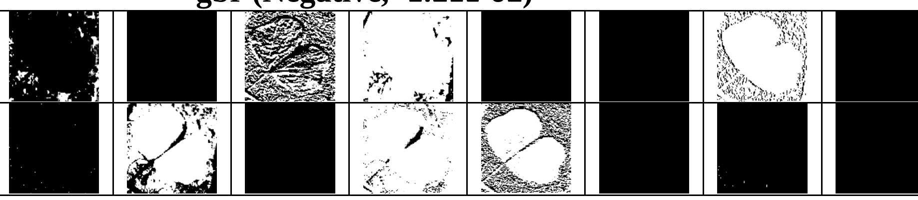
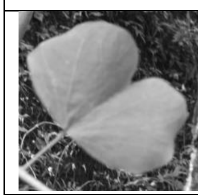
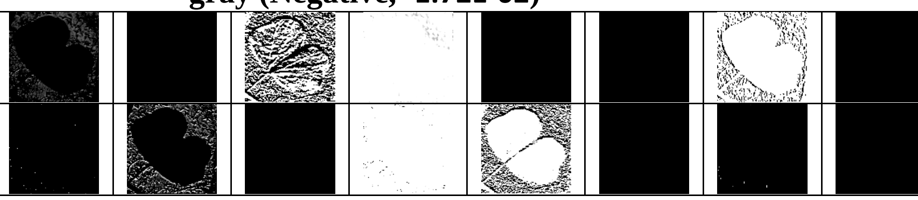
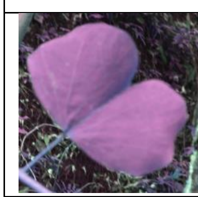
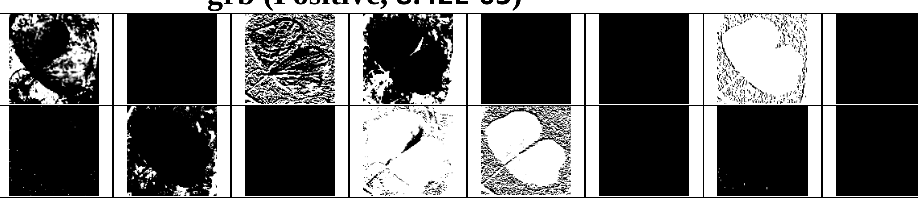
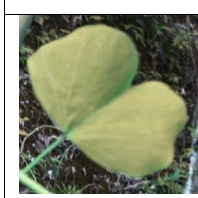
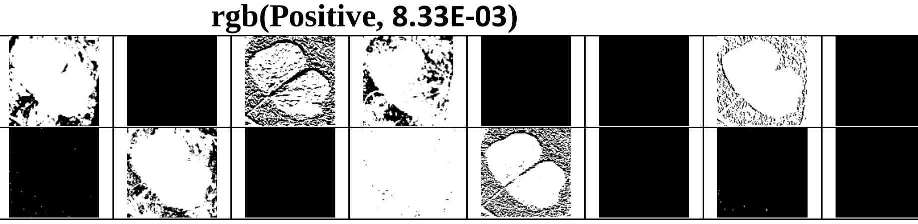
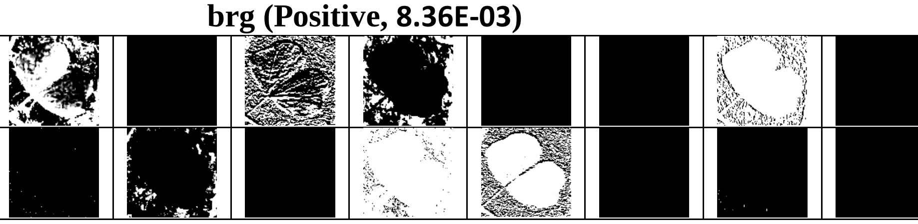
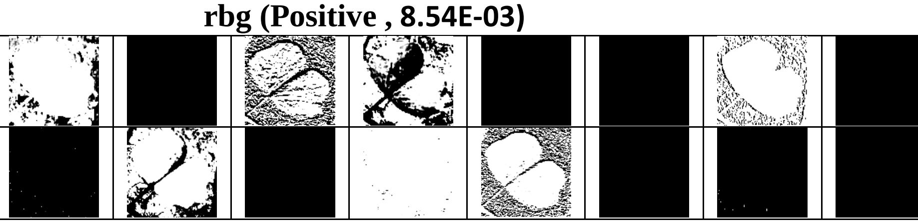
The following table indicates the actual difference prediction score of BV. The detailed information is presented in the appendix 14.4.

Table 23:Summary of prediction score difference of BV in different models

File Name	Score difference against			Difference in average
	model retrained with origin dataset	model retrained with dataset after 1st augmentation	model retrained with dataset after 2nd augmentation	
bv00011_color_aug_bgr.jpg	-1E-05	-0.00014	-6E-05	-7.00E-05
bv00011_color_aug_brg.jpg	0.00431	0.01651	0.00425	8.36E-03
bv00011_color_aug_gbr.jpg	-0.00399	-0.02291	-0.00929	-1.21E-02
bv00011_color_aug_gray.jpg	-0.03866	-0.00876	-0.00426	-1.72E-02
bv00011_color_aug_grb.jpg	0.00421	0.01654	0.00451	8.42E-03
bv00011_color_aug_rbg.jpg	0.00441	0.01672	0.0045	8.54E-03
bv00011_color_aug_rgb.jpg	0.00437	0.01612	0.0045	8.33E-03

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*

Table 24: Summary of output channels in 0-th layer of different color-changed leaves

Input	Output in 0-th layer
bgr (n/a)	
	
gbr (Negative, -1.21E-02)	
	
gray (Negative, -1.72E-02)	
	
grb (Positive, 8.42E-03)	
	
rgb(Positive, 8.33E-03)	
	
brg (Positive, 8.36E-03)	
	
rbg (Positive , 8.54E-03)	
	

In the above table, output channel in the 0-th layer from different color changing methods are shown. Positive means the improvement exists in the recognition, followed by the score difference. Negative means the regression exists in the recognition, followed by the score difference.

According to the table above, no matter how the color changes, the same of activated channel is found. Channel 2, 5, 6, 8, 9, 11, 14, 15, 16 are still totally in black among color-changed images. The rest of the channels are the visible activation of learned features.

As for the images with negative result, we can observe that an obvious difference is found channel 1 and 4. We can inferred that the particular color changing method may possibly cause the loss of information and less activation of output channel occurs. The missing information will eventually influence the classification.

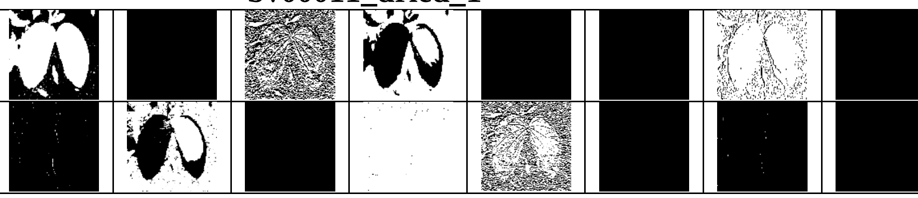
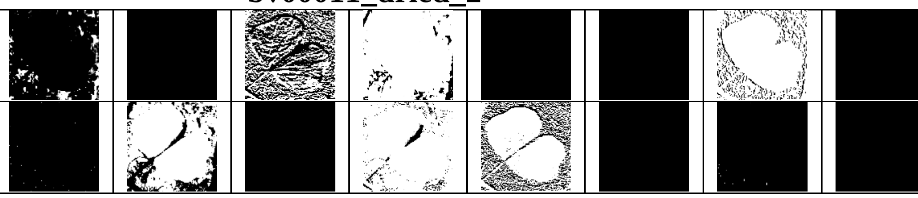
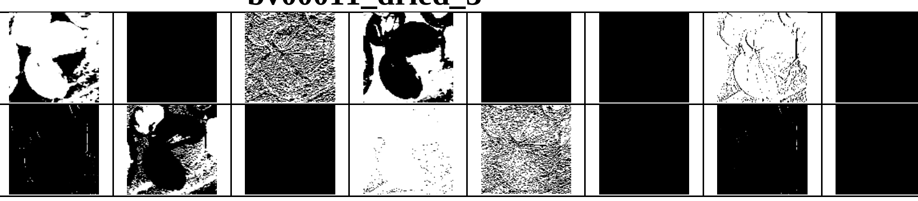
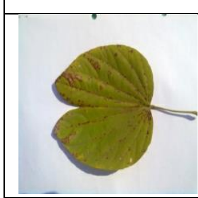
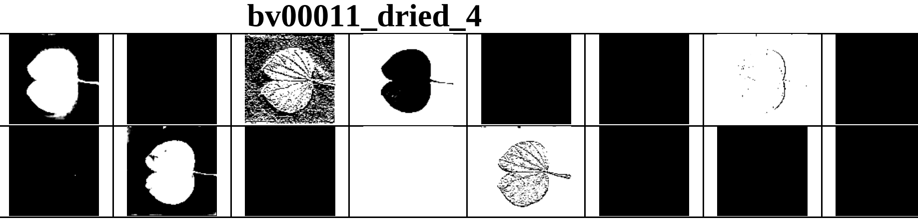
Despite of the score deduction, the model is found that classification still works for the leaves. One of the reasons is that shape and contour are still be extracted from the input image, according to the shown outputs. Therefore, color changing may not have the significant effect toward recognition and our application may still handle most of the cases.

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*

7.5.5. Visualization regarding dried leaves

As for the dried leaves, if the shape of the leaves and most of the leaves area are preserved, the model should be able to learn and recognizing the leaves. The following table only shows the outputs in 0-th layer, see appendix 14.6 for detailed visualizations.

Table 25: Summary of output channels in 0-th layer of different dried leaves

Input	**Outputs in 0-th layer**
bv00011_dried_1	
	
bv00011_dried_2	
	
bv00011_dried_3	
	
bv00011_dried_4	
	

In table Table 25, despite of the variation of color and broken area, the model can still tell the shape of the leaves. For example, in channel 1, 4, 10 and 13, the shape of the leaves has been activated in the convolution operation. Therefore, our application should still work with dried leaves, if the special shape and area can still be observed in reality.

8. Limitation and difficulties

8.1. Data Collection

To obtain a large amount of labelled data, finding existed datasets and crawling in the internet are the ideal methods. However, since we only want to investigate the trees in CUHK, we cannot find the tree dataset we needed. For crawling images in the internet, we can obtain labelled images by typing keyword on Google Image. However, we cannot get high-quality image data without filtering and the label on the images might be incorrect. Without professional knowledge on trees, we cannot identify the trees easily. By feeding the images labelled incorrectly, we could possibly obtain bad results.

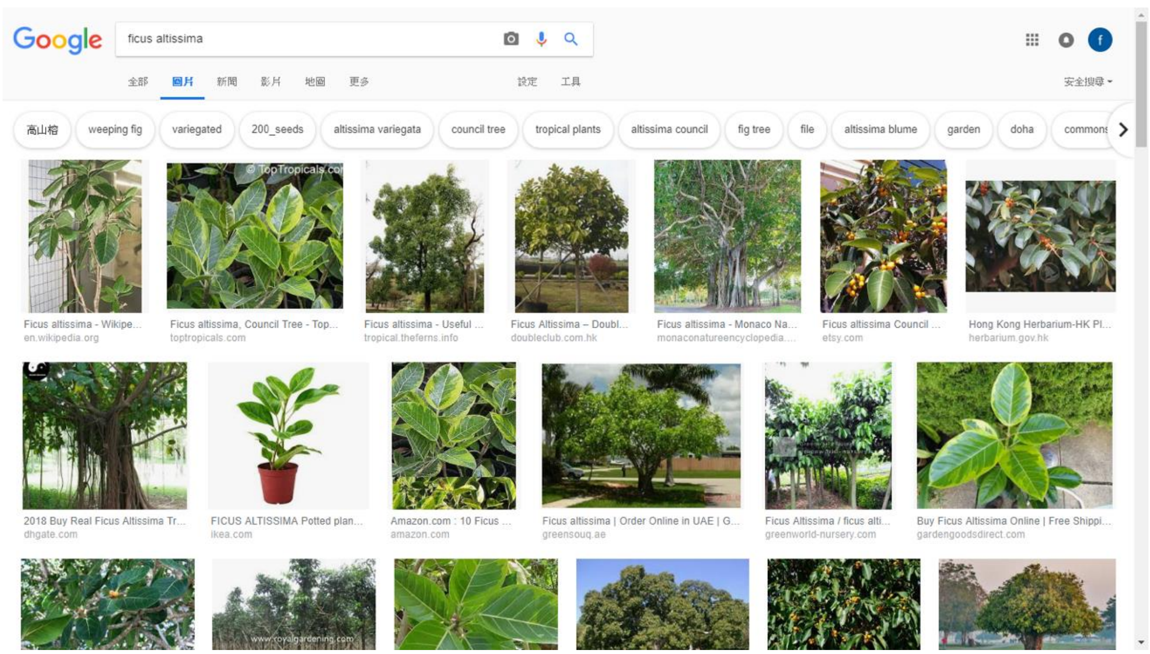


Figure 71: Search results of ficus altissima on Google Image

Some leaves are too high, we cannot pick and take photo with it, therefore it will be the limitation of the variety of plant in dataset if we only use the self-taking photo.

8.2. Knowledge about plants

To develop an advanced tree recognizing application, we should not only focus on the AI technology but also the characteristics of trees.

Before the whole development process, we have make an inquiry to Dr. David Lau, the Curator of Shiu-Ying Hu Herbarium CUHK. Unexpectedly, relying on leaves only is not enough to differentiate the species of trees in professional point of view. For scientific study and forensic evidence, more that 15 features of trees would be used normally. Similar to human being, the features of one species and those of their relatives are very similar. With only a few features, the classification could possibly incorrect.

Figure 72: Flowers of Bauhinia variegate(left) and Hong Kong Orchid Tree(right).[37]

Besides, even for the same species of tree, their features could be not the same if they grow in different environment. With different intensity of sun light or different climate change, their features could vary. Since our data is collected in CUHK only, there could be "overfitting" problem and the application could not be able to recognize the same kind of tree growing in USA.

[37] Picture is retrieved from https://zh.wikipedia.org/wiki/%E6%B4%8B%E7%B4%AB%E8%8D%8A, https://zh.wikipedia.org/wiki/%E5%AE%AE%E7%B2%89%E7%BE%8A%E8%B9%84%E7%94%B2

9. Division of labor

This project is divided into parts, one is related to mobile application and the other part is focusing on AI model. In the course of this FYP, Edward and Paul took turns to contribute the project in different semester. The following table can summarize the major accomplishments of each of us.

	Mobile Application	AI Model
1st Semester (Sep – Dec 2018)	**<u>PIC: Edward</u>** **1. Dr. Leaf V1.0** 1.1. Market Research 1.2. Initial Design (UI) 1.3. Development (Function) 1.4. Testing	**<u>PIC: Paul</u>** 1. Work on Neural Network 1.1.Residual Net 1.2.MobileNet
2nd Semester (Jan – Apr 2019)	**<u>PIC: Paul</u>** 1. Dr.Leaf V2.0 1.1.Mobile app interface 1.2.Server	**<u>PIC: Edward</u>** 1. Experiment of MobileNet (Chapter 7) 2. Model Visualization

Contribution to the report can be summarized as follows,

Index	**Title**	**Person in Charge**			
		Edward		**Paul**	
0.	Abstract	**X**	**1**		
1.	Introduction	**X**	**2**		
2.	Technology Overview	**X**	**3**	**X**	**1**
3.	Preparation	**X**	**2**		
4.	Specification of Dr.Leaf V1.0	**X**	**31**		
5.	Project Details (Specification of Dr. Leaf V2.0)			**X**	**26**
6.	AI Model			**X**	**11**
7.	Experiment and Investigation	**X**	**36**		
8.	Limitation and difficulties			**X**	**2**
9.	Division of labor	**-**	**-**	**-**	**-**
10.	Conclusion	**X**	**1**	**X**	**1**
11.	Future Development	**X**	**1**	**X**	**1**
12.	Acknowledgement	**X**	**1**	**X**	**1**
13.	References	**-**	**-**	**-**	**-**
14.	Appendix	**X**	**24**		
	Total Page Count	**102**		**43**	

10. Conclusion

10.2. Mobile Application

An auto-training module is added for users to contribute to the dataset, in order to optimize the AI model. The module is separated into two parts, which are the mobile app and the server, due to the limited storage and computing power of mobile device.

The whole module is developed in a bottom-up manner. Basic units are tested before composing. This reduces the potential error in the development.

This module only provides users a convenient way to collect the dataset but not ensure that the re-trained model will be always better than the older one. The quality of the recognizing model still relies on the quality of the new incoming image data and the difficulty of recognizing certain kind of plants. This module is more or less providing a convenient way to find out which species are more difficult to be recognized through neural network.

10.3. AI Model

Four experiments have been established in order to improve the recognition from the ground of retraining a better model and carry out the test for the tricky case in the reality. According to the result, (i) it is proved that data augmentation regarding affine transformation can improve the recognition as it simulates the situation of taking photo from different angles in real user case and it resolve the problem of distortion in camera. (ii) Data augmentation regarding lighting condition can be implemented with a wider range of lighting variables since it is found that more information and knowledge is provider for the classifier in the retrained-model to classify a particular specie of leaves. (iii) Despite of the deduction in prediction score of color-changed images (leaves), the model can still recognize the leaves with a lower score by the shape and contour. (iv) Dried leaves with various color found on the surface and some damaged area are still able to recognize if the shape is preserved. However, failure of classification may occur for the leaves which may not have a special shape, or another similar shape of leaf exists in the database.

Therefore, the application is developed and built with support of research-based evidence and we might hope to able contribute the similar application in the community with the above experiments.

11. Future Development

(or say leaving some question for people who would like to pick this topic up in the future)

11.2. Mobile Application

In a short development period, many important parts are not implemented. If we want to put the application into practical use, we need to consider more about the real situation.

In real situation, number of users could be large. When two requests come to the server at the same time, conflict may happen. Therefore, a request queue must be made to handle the request one by one.

Besides, for a large number of users, a single machine is not able to handle a million of request. Therefore, multiple number of servers must be set. Distributed storage and distributed computing could also be considered.

11.3. AI Model

As the project is set up with sustainability, further research and investigation can be initiated to contribute the topic of AI recognition in leaves.

First of all, a larger scale of database can be adopted and extend the application to support more kinds of leaves rather than plants only in CUHK. The movement could contribute to the research of AI recognition and the community in terms of realistic application in daily life.

Moreover, in the experiment of data augmentation regarding lighting conditions, the method of performance can still be investigated in the future. For instance, in the paper discussing facial recognition, lighting augmentation can be performed based on the 3d reconstruction of face shape from the 2d image. [9] If we can share the same method in leaves recognition, we might a better method to simulate lighting from different angles and the augmentation should be closer to reality.

As for the experiment of recognizing different color-changed images, one of the future investigations can review the method color conversion with less or even zero information loss as mentioned. Furthermore, augmentation regarding changing color of the image can be studied if the color conversion method can produce a different information or knowledge to the classifier with a less information loss.

12. Acknowledgement

We would like to express our sincere and deep gratitude to our final year project supervisor Prof. Lyu Rung Tsong Michael and our tutor Xu hui for providing us the assistance regarding ideation as well as technical knowledge to overcome the difficulties we encounter on the road of the project, by directing us to the right path accomplishing the milestones and missions. Moreover, we would like to give credit to Dr.Tai Wai David Lau, the Curator of Shiu-Ying Hu Herbarium CUHK. Without the guidance from him on the knowledge of plants and the suggestion of make our own dataset, we might not able to locate correctly the plants in CUHK and struggle to get the raw data by ourselves.

13. References

[1] Mansurov, N. M. (n.d.). What is Distortion? Retrieved from https://photographylife.com/what-is-distortion

[2] Image Preprocessing. (n.d.). Retrieved from https://keras.io/preprocessing/image/

[3] Kanan C, Cottrell GW (2012) Color-to-Grayscale: Does the Method Matter in Image Recognition? PLoS ONE 7(1): e29740. doi:10.1371/ journal.pone.0029740

[4] Color conversions. (n.d.). Retrieved from https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html

[5] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). springer, Cham

[6] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

[7] Krizhevsky, A., & Hinton, G. (2010). Convolutional deep belief networks on cifar-10. Unpublished manuscript, 40(7).

[8] Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8*(6), 679-698.

[9] Crispell, D., Biris, O., Crosswhite, N., Byrne, J., & Mundy, J. L. (2017). Dataset augmentation for pose and lighting invariant face recognition. arXiv preprint arXiv:1704.04326.

[10] Jeon, W. S., & Rhee, S. Y. (2017). Plant leaf recognition using a convolution neural network. International Journal of Fuzzy Logic and Intelligent Systems, 17(1), 26-34.

[11] Activities | Android Developers. (n.d.). Retrieved from https://developer.android.com/guide/components/activities/

[12] CardView | Android Developers. (n.d.). Retrieved from https://developer.android.com/reference/android/support/v7/widget/CardView

[13] CUHK campus 100 plants| Shiu Ying Hu Herbarium Collections. (n.d.). Retrieved from http://syhuherbarium.sls.cuhk.edu.hk/collections/courseware/campus-100-plants/

[14] Ghezzi, C., Jazayeri, M., & Mandrioli, D. (2003). Fundamentals of software engineering (2nd ed.). Upper Saddle River, N.J.: Prentice Hall.

[15] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, 770-778.

[16] Hornik, Stinchcombe, & White. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359-366.

[17] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

[18] IDC - Smartphone Market Share - OS. (n.d.). Retrieved November 26, 2018, from https://www.idc.com/promo/smartphone-market-share/os

[19] Save data in a local database using Room | Android Developers. (n.d.). Retrieved from https://developer.android.com/training/data-storage/room/

[20] Smit, R. (n.d.). Blog | The Intent extras size limit. Retrieved from https://www.neotechsoftware.com/blog/android-intent-size-limit

[21] Stack Overflow - Where Developers Learn, Share, & Build Careers. (n.d.). Retrieved from https://stackoverflow.com/

[22] Szegedy, Wei Liu, Yangqing Jia, Sermanet, Reed, Anguelov, . . . Rabinovich. (2015). Going deeper with convolutions. In CVPR, 2015.

[23] Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

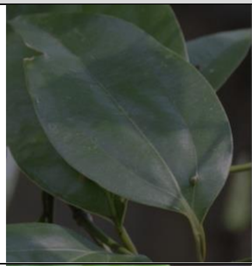
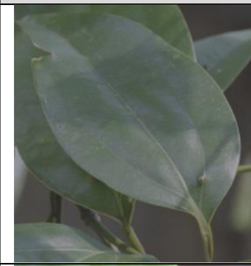
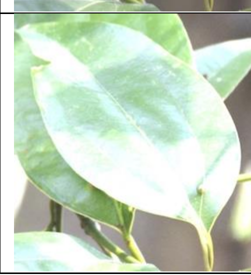
14. Appendix

14.1. List of Figures

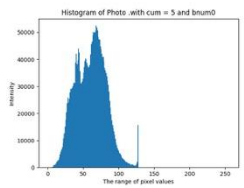
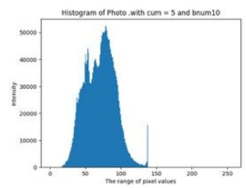
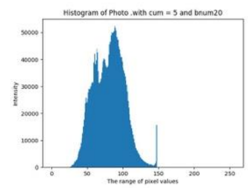
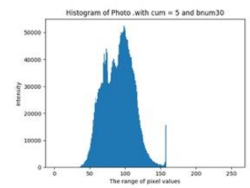
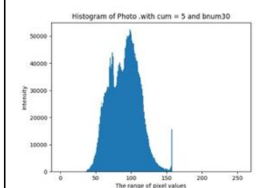
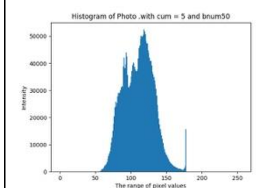
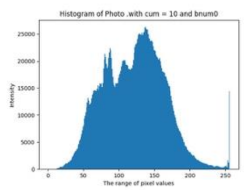
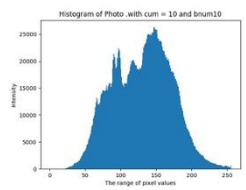
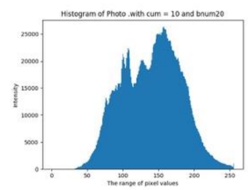
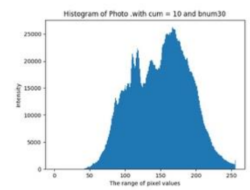
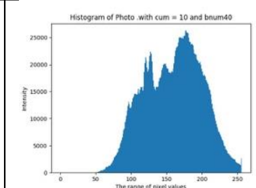
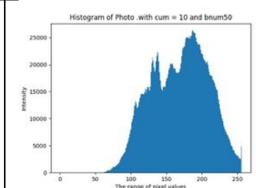
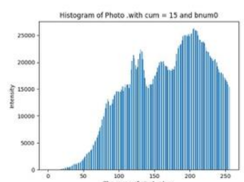
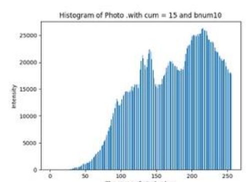
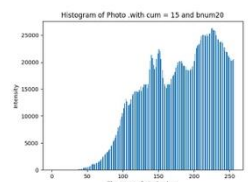
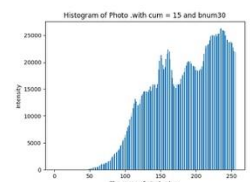
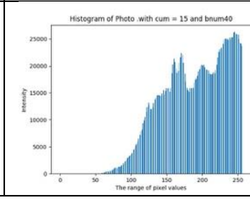
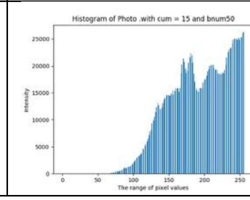
14.2. List of Tables

14.3. Experiment II

14.3.1. Image demonstration of different lighting variables

bnum / cnum	0	10	20	30	40	50
5						
10						
15						

14.3.2. Image demonstration of histograms of different lighting variables

bnum / cnum	0	10	20	30	40	50
5						
10						
15						

14.3.3. Images used for testing

Filename is formatted as "{label}_{bright/dark}.jpg".

Label	***Bright***	***Dark***
BV		
CA		
CB		
CG		
CJ		
FA		

HC		
HR		
MC		
SL		

14.3.4. Prediction with model retrained with origin dataset

	Top 1		Top 2		Top 3		Top 4		Top 5	
File name	label	score	label	score	label	score	label	score	label	score
bv_bright.jpg	bv	0.99871	hc	0.00116	fa	0.00006	cj	0.00003	hr	0.00002
bv_dark.jpg	bv	0.9945	sl	0.00512	fa	0.00023	cb	0.00015	cj	0
ca_bright.jpg	hc	0.8845	ca	0.06728	hr	0.02682	sl	0.00978	fa	0.00491
ca_dark.jpg	ca	0.87744	cj	0.09154	cg	0.015	fa	0.01156	cb	0.00298
cb_bright.jpg	cb	0.99091	hc	0.00299	fa	0.00189	cj	0.00163	sl	0.00102
cb_dark.jpg	cb	0.99992	cj	0.00004	mc	0.00002	fa	0.00001	sl	0.00001
cg_bright.jpg	cg	0.98245	hc	0.01551	fa	0.00179	sl	0.00017	ca	0.00005
cg_dark.jpg	cg	0.99968	ca	0.0003	hc	0.00002	sl	0	fa	0
cj_bright.jpg	cj	0.99985	cb	0.00008	fa	0.00005	bv	0.00002	sl	0
cj_dark.jpg	cj	0.99816	fa	0.00141	cb	0.00042	hc	0.00001	cg	0
fa_bright.jpg	fa	0.98338	bv	0.01567	sl	0.00036	cj	0.00032	cg	0.00017
fa_dark.jpg	fa	0.86376	bv	0.13611	cg	0.00008	cj	0.00003	ca	0.00002
hc_bright.jpg	hc	0.99882	sl	0.00064	cg	0.00036	cb	0.00008	cj	0.00007
hc_dark.jpg	hc	0.99733	cg	0.00267	ca	0	fa	0	cb	0
hr_bright.jpg	hr	0.98133	bv	0.01132	hc	0.00721	sl	0.00013	cj	0
hr_dark.jpg	hr	0.99923	hc	0.00073	cj	0.00003	bv	0.00001	fa	0
mc_bright.jpg	mc	0.99258	bv	0.00655	fa	0.00083	sl	0.00002	cb	0.00002
mc_dark.jpg	mc	0.96526	sl	0.02287	fa	0.00826	cb	0.00241	cg	0.00052
sl_bright.jpg	sl	0.92493	ca	0.07185	fa	0.00212	cg	0.00063	hc	0.0003
sl_dark.jpg	sl	0.87538	cg	0.10225	ca	0.0166	hc	0.00358	cb	0.00214

14.3.5. Prediction with model retrained with 1st augmented data

	Top 1		Top 2		Top 3		Top 4		Top 5	
File name	label	score	label	score	label	score	label	score	label	score
bv_bright.jpg	bv	0.99952	hr	0.00027	fa	0.00012	hc	0.00008	sl	0
bv_dark.jpg	bv	0.99715	sl	0.00143	fa	0.00076	cb	0.00064	cj	0.00002
ca_bright.jpg	hc	0.71984	ca	0.24871	cg	0.01123	sl	0.01058	cj	0.00619
ca_dark.jpg	cg	0.51249	ca	0.3984	fa	0.06021	cb	0.01196	cj	0.01062
cb_bright.jpg	cb	0.95536	cj	0.01496	hc	0.0128	sl	0.00992	hr	0.00555
cb_dark.jpg	cb	0.99997	sl	0.00002	cj	0.00001	mc	0	fa	0
cg_bright.jpg	cg	0.98583	hc	0.00865	fa	0.00488	sl	0.00053	ca	0.00008
cg_dark.jpg	cg	0.99975	ca	0.00024	hc	0.00001	sl	0	fa	0
cj_bright.jpg	cj	0.9998	cb	0.00017	bv	0.00001	fa	0.00001	hc	0.00001
cj_dark.jpg	cj	0.99809	cb	0.00143	fa	0.00045	hc	0.00002	cg	0.00001
fa_bright.jpg	fa	0.97914	bv	0.02012	cj	0.00048	cg	0.00021	cb	0.00004
fa_dark.jpg	fa	0.86202	bv	0.13792	cg	0.00003	ca	0.00003	cj	0
hc_bright.jpg	hc	0.99972	sl	0.00021	cb	0.00004	cg	0.00002	cj	0.00001
hc_dark.jpg	hc	0.99984	cg	0.00014	ca	0.00001	sl	0	fa	0
hr_bright.jpg	hr	0.9924	bv	0.00658	hc	0.00099	sl	0.00003	cj	0
hr_dark.jpg	hr	0.99737	hc	0.00224	cj	0.00027	bv	0.00011	fa	0.00001
mc_bright.jpg	mc	0.94465	bv	0.03819	fa	0.01624	cb	0.00055	sl	0.00037
mc_dark.jpg	mc	0.48928	sl	0.40143	cg	0.05117	cb	0.04416	fa	0.00722
sl_bright.jpg	sl	0.87673	ca	0.11655	hc	0.00258	fa	0.00195	cg	0.00194
sl_dark.jpg	sl	0.86204	cg	0.07795	ca	0.05659	hc	0.00209	cb	0.00128

14.3.6. Prediction with model retrained with augmented data according the lighting variables of SET 1

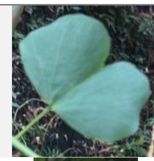
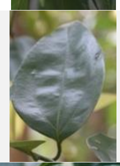
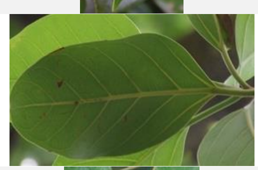
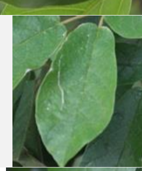
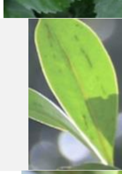
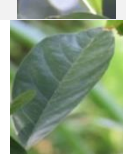
	Top 1		Top 2		Top 3		Top 4		Top 5	
File name	label	score	label	score	label	score	label	score	label	score
bv_bright.jpg	bv	0.99925	fa	0.00072	hc	0.00002	hr	0.00001	cj	0
bv_dark.jpg	bv	0.99834	fa	0.00104	sl	0.00048	cb	0.00013	cj	0.00001
ca_bright.jpg	ca	0.5896	hc	0.3128	sl	0.07048	cj	0.01534	cg	0.0091
ca_dark.jpg	ca	0.59977	cg	0.34516	fa	0.0242	cj	0.0179	cb	0.00995
cb_bright.jpg	cb	0.88185	sl	0.05562	cj	0.05186	hr	0.0053	hc	0.00368
cb_dark.jpg	cb	0.99972	cj	0.00015	sl	0.00009	mc	0.00004	fa	0
cg_bright.jpg	cg	0.99535	hc	0.00225	fa	0.00162	sl	0.00062	ca	0.00013
cg_dark.jpg	cg	0.99981	ca	0.00019	hc	0	fa	0	sl	0
cj_bright.jpg	cj	0.99997	cb	0.00002	bv	0	fa	0	sl	0
cj_dark.jpg	cj	0.99906	cb	0.00077	fa	0.00014	hc	0.00002	cg	0.00001
fa_bright.jpg	fa	0.98521	bv	0.01125	cj	0.00284	cg	0.00065	hc	0.00002
fa_dark.jpg	fa	0.87345	bv	0.12645	cg	0.00006	ca	0.00003	cj	0.00001
hc_bright.jpg	hc	0.9958	sl	0.00377	cg	0.0002	cb	0.00011	cj	0.00011
hc_dark.jpg	hc	0.99989	cg	0.0001	fa	0.00001	ca	0	sl	0
hr_bright.jpg	hr	0.91952	bv	0.06882	hc	0.00865	sl	0.00288	cj	0.00011
hr_dark.jpg	hr	0.99214	hc	0.00735	cj	0.00032	bv	0.00018	fa	0.00001
mc_bright.jpg	mc	0.97052	bv	0.02044	fa	0.00839	sl	0.00053	cb	0.00012
mc_dark.jpg	sl	0.5244	mc	0.32452	cg	0.07087	cb	0.05762	ca	0.01388
sl_bright.jpg	sl	0.95879	ca	0.03965	cg	0.00088	fa	0.00029	hc	0.00022
sl_dark.jpg	sl	0.93595	cg	0.03672	ca	0.02649	cb	0.0005	hc	0.00028

14.3.7. Prediction with model retrained with augmented data according the lighting variables of SET 2

	Top 1		Top 2		Top 3		Top 4		Top 5	
File name	label	score	label	score	label	score	label	score	label	score
bv_bright.jpg	bv	0.99987	fa	0.00007	hc	0.00003	hr	0.00003	cj	0
bv_dark.jpg	bv	0.99957	fa	0.00022	sl	0.00011	cb	0.0001	cj	0
ca_bright.jpg	ca	0.5528	hc	0.39101	cg	0.03562	sl	0.01625	cj	0.00158
ca_dark.jpg	cg	0.55116	ca	0.43227	cb	0.00793	fa	0.00476	cj	0.00212
cb_bright.jpg	cb	0.93211	sl	0.02705	cj	0.01823	hc	0.01113	hr	0.00785
cb_dark.jpg	cb	0.99996	sl	0.00002	mc	0.00002	cj	0.00001	fa	0
cg_bright.jpg	cg	0.99438	fa	0.00328	hc	0.00209	ca	0.00017	sl	0.00007
cg_dark.jpg	cg	0.99973	ca	0.00027	hc	0	fa	0	sl	0
cj_bright.jpg	cj	0.99982	cb	0.00015	bv	0.00002	hr	0	fa	0
cj_dark.jpg	cj	0.99671	cb	0.00306	fa	0.00016	cg	0.00005	hc	0.00001
fa_bright.jpg	fa	0.97855	bv	0.0187	cj	0.00151	cg	0.00116	cb	0.00004
fa_dark.jpg	fa	0.54897	bv	0.45037	ca	0.00041	cg	0.00024	cj	0.00001
hc_bright.jpg	hc	0.99956	sl	0.00026	cg	0.00012	cb	0.00005	cj	0.00001
hc_dark.jpg	hc	0.99953	cg	0.00045	fa	0.00001	ca	0.00001	sl	0
hr_bright.jpg	hr	0.88863	bv	0.11007	hc	0.00122	sl	0.00008	cj	0
hr_dark.jpg	hr	0.99895	hc	0.00089	bv	0.00013	cj	0.00003	fa	0
mc_bright.jpg	mc	0.95077	bv	0.0461	fa	0.00297	sl	0.00008	cb	0.00008
mc_dark.jpg	mc	0.71835	sl	0.17296	cb	0.05263	cg	0.043	fa	0.00764
sl_bright.jpg	sl	0.50232	ca	0.49177	cg	0.00411	fa	0.00102	hc	0.00061
sl_dark.jpg	sl	0.48486	cg	0.30027	ca	0.21181	cb	0.00193	hc	0.00105

14.4. Experiment III

14.4.1. 10 leaves for Experiment III

LABEL	PHOTO
BV	
CA	
CB	
CG	
CJ	
FA	
HC	
HR	
MC	
SL	

14.4.2. Prediction with model retrained with origin dataset

	Top 1		Top 2		Top 3		Top 4		Top 5		
File Name	Label	Score	Label	Score	Label	Score	Label	Score	Label	Score	Prediction score compare to origin photo
bv00011_color_aug_bgr.jpg	bv	0.99552	cg	0.00412	sl	0.00018	hc	0.00009	cj	0.00005	-1E-05
bv00011_color_aug_brg.jpg	bv	0.99984	cj	0.00014	cg	0.00002	mc	0	sl	0	0.00431
bv00011_color_aug_gbr.jpg	bv	0.99154	cg	0.00678	sl	0.00107	hc	0.00027	cj	0.00021	-0.00399
bv00011_color_aug_gray.jpg	bv	0.95687	cg	0.03291	cj	0.00846	sl	0.00091	hc	0.00043	-0.03866
bv00011_color_aug_grb.jpg	bv	0.99974	cg	0.00022	hc	0.00001	sl	0.00001	mc	0.00001	0.00421
bv00011_color_aug_rbg.jpg	bv	0.99994	cj	0.00004	cg	0.00002	hc	0	mc	0	0.00441
bv00011_color_aug_rgb.jpg	bv	0.9999	cj	0.00005	cg	0.00004	hc	0	mc	0	0.00437
bv00011.jpg	bv	0.99553	cg	0.00413	sl	0.00018	hc	0.00007	cj	0.00006	
ca00003_color_aug_bgr.jpg	ca	0.99931	fa	0.00024	cg	0.00022	sl	0.00019	cj	0.00003	8E-05
ca00003_color_aug_brg.jpg	ca	0.99803	sl	0.00109	cj	0.00041	cg	0.00028	fa	0.00019	-0.0012
ca00003_color_aug_gbr.jpg	ca	0.9944	sl	0.00538	cg	0.0002	cj	0.00001	fa	0	-0.00483
ca00003_color_aug_gray.jpg	ca	0.99965	sl	0.00029	fa	0.00004	cg	0.00002	cj	0.00001	0.00042
ca00003_color_aug_grb.jpg	ca	0.99969	sl	0.00029	cg	0.00002	fa	0	cj	0	0.00046
ca00003_color_aug_rbg.jpg	ca	0.99047	sl	0.00458	cj	0.0025	cg	0.00144	fa	0.0009	-0.00876
ca00003_color_aug_rgb.jpg	ca	0.99982	sl	0.00008	cg	0.00006	cj	0.00002	fa	0.00001	0.00059
ca00003.jpg	ca	0.99923	fa	0.0003	cg	0.00026	sl	0.00017	cj	0.00003	
cb00009_color_aug_bgr.jpg	cb	0.99701	sl	0.00286	fa	0.00006	cg	0.00005	mc	0.00002	0.00029
cb00009_color_aug_brg.jpg	cb	0.99908	sl	0.0008	fa	0.00008	cg	0.00003	mc	0.00001	0.00236
cb00009_color_aug_gbr.jpg	cb	0.99176	sl	0.00698	cg	0.00119	mc	0.00006	fa	0.00001	-0.00496
cb00009_color_aug_gray.jpg	sl	0.66417	cb	0.32186	fa	0.00937	mc	0.00296	cj	0.00135	-0.67486
cb00009_color_aug_grb.jpg	cb	0.95553	sl	0.03848	cg	0.00593	mc	0.00003	hc	0.00001	-0.04119
cb00009_color_aug_rbg.jpg	cb	0.99432	sl	0.00525	cg	0.0003	fa	0.00007	cj	0.00003	-0.0024
cb00009_color_aug_rgb.jpg	cb	0.98409	sl	0.01473	cg	0.00102	hc	0.00005	mc	0.00005	-0.01263
cb00009.jpg	cb	0.99672	sl	0.00314	fa	0.00006	cg	0.00005	mc	0.00002	
cg00030_color_aug_bgr.jpg	cg	0.9999	hc	0.00009	ca	0.00001	sl	0	fa	0	2E-05
cg00030_color_aug_brg.jpg	cg	0.99852	hc	0.0014	ca	0.00006	sl	0.00002	fa	0	-0.00136
cg00030_color_aug_gbr.jpg	cg	0.99985	ca	0.00013	sl	0.00001	hc	0.00001	fa	0	-3E-05
cg00030_color_aug_gray.jpg	cg	0.99993	sl	0.00005	ca	0.00002	hc	0	fa	0	5E-05
cg00030_color_aug_grb.jpg	cg	0.99962	ca	0.00036	sl	0.00001	hc	0.00001	fa	0	-0.00026
cg00030_color_aug_rbg.jpg	cg	0.99954	hc	0.00044	ca	0.00001	sl	0.00001	fa	0	-0.00034
cg00030_color_aug_rgb.jpg	cg	0.99988	hc	0.00009	ca	0.00003	fa	0	sl	0	0
cg00030.jpg	cg	0.9999	hc	0.00009	ca	0.00001	sl	0	fa	0	
cj00010_color_aug_bgr.jpg	cj	0.99994	fa	0.00004	sl	0.00002	hc	0	cb	0	0
cj00010_color_aug_brg.jpg	cj	0.99999	sl	0.00001	fa	0	hc	0	cb	0	5E-05
cj00010_color_aug_gbr.jpg	cg	0.44977	fa	0.22323	sl	0.13698	cj	0.09722	ca	0.09104	-0.90272
cj00010_color_aug_gray.jpg	cj	0.99559	fa	0.00331	sl	0.00106	ca	0.00002	bv	0.00001	-0.00435
cj00010_color_aug_grb.jpg	cg	0.56403	ca	0.23826	cj	0.12239	sl	0.0508	fa	0.02086	-0.87755
cj00010_color_aug_rbg.jpg	cj	0.99997	sl	0.00001	fa	0.00001	cg	0	hc	0	3E-05
cj00010_color_aug_rgb.jpg	cj	0.73586	fa	0.17347	cg	0.05219	sl	0.02243	ca	0.01361	-0.26408
cj00010.jpg	cj	0.99994	fa	0.00004	sl	0.00002	hc	0	cb	0	
fa00014_color_aug_bgr.jpg	fa	0.9991	ca	0.00064	cg	0.00014	sl	0.00009	bv	0.00001	-6E-05
fa00014_color_aug_brg.jpg	fa	0.99939	ca	0.00044	cg	0.00008	sl	0.00005	bv	0.00003	0.00023
fa00014_color_aug_gbr.jpg	fa	0.84058	ca	0.14866	sl	0.00684	cg	0.00356	cb	0.0003	-0.15858
fa00014_color_aug_gray.jpg	fa	0.93571	ca	0.05014	sl	0.01314	cg	0.0008	cb	0.00017	-0.06345
fa00014_color_aug_grb.jpg	fa	0.82867	ca	0.15551	sl	0.01167	cg	0.00297	cb	0.00106	-0.17049
fa00014_color_aug_rbg.jpg	fa	0.9988	sl	0.00053	ca	0.00044	bv	0.00015	cb	0.00006	-0.00036
fa00014_color_aug_rgb.jpg	fa	0.86757	ca	0.11511	sl	0.01327	cg	0.00319	cb	0.00054	-0.13159
fa00014.jpg	fa	0.99916	ca	0.00059	cg	0.00013	sl	0.00009	bv	0.00001	
hc00033_color_aug_bgr.jpg	hc	0.99981	cb	0.00016	cj	0.00002	sl	0	hr	0	0
hc00033_color_aug_brg.jpg	hc	0.99967	cb	0.00029	cj	0.00002	sl	0.00001	bv	0	-0.00014
hc00033_color_aug_gbr.jpg	hc	0.70882	cb	0.28794	sl	0.00185	cg	0.00127	fa	0.00006	-0.29099
hc00033_color_aug_gray.jpg	hc	0.59391	cb	0.40388	cg	0.00134	sl	0.00056	bv	0.00025	-0.4059
hc00033_color_aug_grb.jpg	hc	0.88423	cb	0.11474	sl	0.00052	cg	0.00045	bv	0.00005	-0.11558
hc00033_color_aug_rbg.jpg	hc	0.99956	cb	0.00033	cj	0.00005	bv	0.00004	sl	0.00001	-0.00025
hc00033_color_aug_rgb.jpg	hc	0.99999	cb	0.00001	cg	0	sl	0	hr	0	0.00018
hc00033.jpg	hc	0.99981	cb	0.00016	cj	0.00002	sl	0	hr	0	
hr00035_color_aug_bgr.jpg	hr	0.99999	hc	0.00001	bv	0	cj	0	fa	0	0
hr00035_color_aug_brg.jpg	hr	1	hc	0	cj	0	bv	0	sl	0	1E-05
hr00035_color_aug_gbr.jpg	hr	0.99851	hc	0.00134	bv	0.00012	sl	0.00001	cb	0.00001	-0.00148
hr00035_color_aug_gray.jpg	hr	0.99956	hc	0.0004	bv	0.00004	cj	0	fa	0	-0.00043
hr00035_color_aug_grb.jpg	hr	0.99863	hc	0.00131	bv	0.00003	sl	0.00002	ca	0.00001	-0.00136
hr00035_color_aug_rbg.jpg	hr	1	hc	0	bv	0	cj	0	sl	0	1E-05
hr00035_color_aug_rgb.jpg	hr	0.99969	hc	0.0003	bv	0.00001	fa	0	sl	0	-0.0003
hr00035.jpg	hr	0.99999	hc	0.00001	bv	0	cj	0	sl	0	
mc00185_color_aug_bgr.jpg	mc	0.99992	sl	0.00004	cb	0.00003	fa	0	hc	0	-2E-05
mc00185_color_aug_brg.jpg	mc	0.99987	sl	0.00009	cb	0.00003	fa	0	hc	0	-7E-05
mc00185_color_aug_gbr.jpg	mc	0.99612	sl	0.00278	cb	0.00106	fa	0.00003	ca	0	-0.00382
mc00185_color_aug_gray.jpg	mc	0.96701	cb	0.02604	sl	0.00659	fa	0.00035	hc	0	-0.03293
mc00185_color_aug_grb.jpg	mc	0.97077	sl	0.01729	cb	0.01184	fa	0.00007	cg	0.00002	-0.02917
mc00185_color_aug_rbg.jpg	mc	0.99975	sl	0.00023	cb	0.00002	fa	0	hc	0	-0.00019

mc00185_color_aug_rgb.jpg	mc	0.97471	cb	0.01878	sl	0.00646	fa	0.00005	hc	0	-0.02523
mc00185.jpg	mc	0.99994	sl	0.00003	cb	0.00003	fa	0	hc	0	
sl00175_color_aug_bgr.jpg	sl	0.99461	cg	0.00371	ca	0.00105	hc	0.00032	cb	0.00021	-0.00018
sl00175_color_aug_brg.jpg	sl	0.99764	cg	0.00163	hc	0.00035	cb	0.00027	ca	0.00007	0.00285
sl00175_color_aug_gbr.jpg	sl	0.99778	cg	0.00139	ca	0.00077	cb	0.00003	mc	0.00002	0.00299
sl00175_color_aug_gray.jpg	sl	0.94657	cg	0.05079	ca	0.00203	fa	0.00016	mc	0.00013	-0.04822
sl00175_color_aug_grb.jpg	sl	0.98482	cg	0.01114	ca	0.00389	cb	0.00008	mc	0.00004	-0.00997
sl00175_color_aug_rbg.jpg	sl	0.99719	cg	0.00231	cb	0.00019	hc	0.00018	ca	0.00008	0.0024
sl00175_color_aug_rgb.jpg	sl	0.98321	cg	0.01335	hc	0.00173	ca	0.00159	cb	0.00006	-0.01158
sl00175.jpg	sl	0.99479	cg	0.00359	ca	0.00106	hc	0.00028	cb	0.0002	0

14.4.3. Prediction with model retrained with dataset after 1st augmentation

	Top 1		Top 2		Top 3		Top 4		Top 5		
File Name	**Label**	**Score**	**Label**	**Score**	**Label**	**Score**	**Label**	**Score**	**Label**	**Score**	**Prediction score compare to origin photo**
bv00011_color_aug_bgr.jpg	bv	0.98263	cg	0.00806	sl	0.0059	hc	0.0014	mc	0.00113	-0.00014
bv00011_color_aug_brg.jpg	bv	0.99928	cj	0.00056	hr	0.00006	sl	0.00003	mc	0.00003	0.01651
bv00011_color_aug_gbr.jpg	bv	0.95986	sl	0.02596	cg	0.00655	hc	0.00268	hr	0.00236	-0.02291
bv00011_color_aug_gray.jpg	bv	0.97401	sl	0.00758	cj	0.00568	mc	0.00449	cg	0.00447	-0.00876
bv00011_color_aug_grb.jpg	bv	0.99931	sl	0.00027	hc	0.00014	hr	0.0001	cg	0.0001	0.01654
bv00011_color_aug_rbg.jpg	bv	0.99949	hr	0.00019	cj	0.00019	hc	0.00006	sl	0.00003	0.01672
bv00011_color_aug_rgb.jpg	bv	0.99889	hr	0.00051	cj	0.00032	hc	0.00016	sl	0.00006	0.01612
bv00011.jpg	bv	0.98277	cg	0.00777	sl	0.00604	mc	0.00132	hc	0.00117	
ca00003_color_aug_bgr.jpg	ca	0.99966	cg	0.00025	fa	0.00008	sl	0.00001	cj	0	1E-04
ca00003_color_aug_brg.jpg	ca	0.9991	fa	0.00041	cg	0.00038	cj	0.00006	sl	0.00005	-0.00046
ca00003_color_aug_gbr.jpg	ca	0.99944	cg	0.00033	sl	0.00022	fa	0.00001	cj	0	-0.00012
ca00003_color_aug_gray.jpg	ca	0.99983	cg	0.00008	fa	0.00005	sl	0.00003	bv	0	0.00027
ca00003_color_aug_grb.jpg	ca	0.99985	cg	0.0001	sl	0.00005	fa	0.00001	cj	0	0.00029
ca00003_color_aug_rbg.jpg	ca	0.998	cg	0.00127	fa	0.00037	cj	0.00024	sl	0.0001	-0.00156
ca00003_color_aug_rgb.jpg	ca	0.99981	cg	0.00016	fa	0.00002	sl	0.00001	hc	0	0.00025
ca00003.jpg	ca	0.99956	cg	0.0003	fa	0.00012	sl	0.00001	cj	0	
cb00009_color_aug_bgr.jpg	cb	0.99474	sl	0.00515	fa	0.00005	cg	0.00004	mc	0.00003	0.00036
cb00009_color_aug_brg.jpg	cb	0.99935	sl	0.0005	fa	0.00012	cg	0.00001	mc	0	0.00497
cb00009_color_aug_gbr.jpg	cb	0.98103	sl	0.01853	cg	0.00028	mc	0.00008	fa	0.00007	-0.01335
cb00009_color_aug_gray.jpg	sl	0.51367	cb	0.46874	fa	0.01368	ca	0.00151	cg	0.00078	-0.52564
cb00009_color_aug_grb.jpg	cb	0.98962	sl	0.0099	cg	0.00028	fa	0.00018	mc	0.00001	-0.00476
cb00009_color_aug_rbg.jpg	cb	0.99493	sl	0.00467	fa	0.00034	cg	0.00006	mc	0	0.00055
cb00009_color_aug_rgb.jpg	cb	0.98813	sl	0.01157	fa	0.00019	cg	0.00009	mc	0.00001	-0.00625
cb00009.jpg	cb	0.99438	sl	0.00551	fa	0.00005	cg	0.00004	mc	0.00003	
cg00030_color_aug_bgr.jpg	cg	0.99884	hc	0.00115	sl	0	ca	0	fa	0	0.00043
cg00030_color_aug_brg.jpg	cg	0.99712	hc	0.00285	sl	0.00002	fa	0.00001	ca	0	-0.00129
cg00030_color_aug_gbr.jpg	cg	0.99994	hc	0.00003	sl	0.00003	ca	0	fa	0	0.00153
cg00030_color_aug_gray.jpg	cg	0.99982	sl	0.00016	hc	0.00001	ca	0.00001	fa	0	0.00141
cg00030_color_aug_grb.jpg	cg	0.99993	fa	0.00002	sl	0.00002	hc	0.00001	ca	0.00001	0.00152
cg00030_color_aug_rbg.jpg	cg	0.99775	hc	0.00222	sl	0.00002	fa	0.00001	ca	0	-0.00066
cg00030_color_aug_rgb.jpg	cg	0.99841	hc	0.00158	fa	0.00001	ca	0	sl	0	0
cg00030.jpg	cg	0.99886	hc	0.00113	sl	0	ca	0	fa	0	
cj00010_color_aug_bgr.jpg	cj	0.99992	fa	0.00006	sl	0.00001	hc	0	cb	0	-1E-05
cj00010_color_aug_brg.jpg	cj	1	fa	0	sl	0	hc	0	bv	0	7E-05
cj00010_color_aug_gbr.jpg	sl	0.46817	cj	0.19263	fa	0.18157	cg	0.09505	ca	0.06184	-0.8073
cj00010_color_aug_gray.jpg	cj	0.99776	fa	0.00157	sl	0.00046	bv	0.00019	ca	0.00002	-0.00217
cj00010_color_aug_grb.jpg	sl	0.55842	cj	0.17198	cg	0.10257	ca	0.08128	fa	0.07741	-0.82795
cj00010_color_aug_rbg.jpg	cj	0.99995	fa	0.00004	sl	0.00001	hc	0	bv	0	2E-05
cj00010_color_aug_rgb.jpg	cj	0.81863	sl	0.07909	fa	0.07538	ca	0.01231	cg	0.00932	-0.1813
cj00010.jpg	cj	0.99993	fa	0.00006	sl	0.00001	hc	0	cb	0	
fa00014_color_aug_bgr.jpg	fa	0.99954	ca	0.00027	cg	0.00008	bv	0.00006	sl	0.00004	-2E-05
fa00014_color_aug_brg.jpg	fa	0.99943	ca	0.00028	bv	0.00013	cg	0.00007	sl	0.00007	-0.00013
fa00014_color_aug_gbr.jpg	fa	0.959	ca	0.03024	cg	0.00649	sl	0.0039	bv	0.00024	-0.04056
fa00014_color_aug_gray.jpg	fa	0.9875	sl	0.0064	ca	0.00562	cg	0.00033	bv	0.00009	-0.01206
fa00014_color_aug_grb.jpg	fa	0.91729	ca	0.06283	sl	0.01022	cg	0.00871	cb	0.00073	-0.08227
fa00014_color_aug_rbg.jpg	fa	0.99938	bv	0.00042	sl	0.0001	ca	0.00007	cb	0.00002	-0.00018
fa00014_color_aug_rgb.jpg	fa	0.87404	ca	0.09267	sl	0.01867	cg	0.01228	bv	0.00142	-0.12552
fa00014.jpg	fa	0.99956	ca	0.00026	cg	0.00007	bv	0.00006	sl	0.00004	
hc00033_color_aug_bgr.jpg	hc	0.99715	cb	0.00282	cj	0.00001	bv	0.00001	hr	0.00001	0
hc00033_color_aug_brg.jpg	hc	0.99304	cb	0.00673	bv	0.00013	hr	0.00005	cj	0.00003	-0.00411
hc00033_color_aug_gbr.jpg	cb	0.82477	hc	0.17336	sl	0.00158	cg	0.00022	bv	0.00005	-0.82379
hc00033_color_aug_gray.jpg	cb	0.96509	hc	0.03333	bv	0.00106	sl	0.00047	cg	0.00004	-0.96382
hc00033_color_aug_grb.jpg	cb	0.74988	hc	0.2494	sl	0.00043	bv	0.00019	cg	0.00008	-0.74775
hc00033_color_aug_rbg.jpg	hc	0.99451	cb	0.00446	bv	0.00078	hr	0.00017	cj	0.00003	-0.00264
hc00033_color_aug_rgb.jpg	hc	0.99991	cb	0.00009	bv	0	hr	0	cg	0	0.00276
hc00033.jpg	hc	0.99715	cb	0.00282	cj	0.00001	hr	0.00001	bv	0.00001	
hr00035_color_aug_bgr.jpg	hr	0.99997	hc	0.00003	bv	0	cj	0	sl	0	0
hr00035_color_aug_brg.jpg	hr	1	hc	0	bv	0	cj	0	fa	0	3E-05

hr00035_color_aug_gbr.jpg	hr	0.99179	hc	0.00667	bv	0.00136	sl	0.00018	cb	0.00001	-0.00818
hr00035_color_aug_gray.jpg	hr	0.99932	bv	0.00052	hc	0.00016	cj	0	sl	0	-0.00065
hr00035_color_aug_grb.jpg	hr	0.99797	hc	0.00146	bv	0.00045	sl	0.00011	cb	0	-0.002
hr00035_color_aug_rbg.jpg	hr	1	bv	0	hc	0	cj	0	sl	0	3E-05
hr00035_color_aug_rgb.jpg	hr	0.99914	hc	0.00082	bv	0.00004	sl	0	cj	0	-0.00083
hr00035.jpg	hr	0.99997	hc	0.00003	bv	0	cj	0	sl	0	
mc00185_color_aug_bgr.jpg	mc	0.99639	sl	0.0018	cb	0.00167	fa	0.00012	hr	0.00001	-0.00037
mc00185_color_aug_brg.jpg	mc	0.99629	sl	0.00302	cb	0.00063	fa	0.00004	hr	0.00002	-0.00047
mc00185_color_aug_gbr.jpg	mc	0.98064	sl	0.01338	cb	0.00542	fa	0.00054	bv	0.00001	-0.01612
mc00185_color_aug_gray.jpg	cb	0.90668	mc	0.05816	sl	0.02956	fa	0.00557	bv	0.00001	-0.9386
mc00185_color_aug_grb.jpg	mc	0.78911	cb	0.12525	sl	0.08528	fa	0.0003	cg	0.00004	-0.20765
mc00185_color_aug_rbg.jpg	mc	0.99776	sl	0.00206	cb	0.00017	fa	0	hr	0	0.001
mc00185_color_aug_rgb.jpg	mc	0.56181	cb	0.37628	sl	0.06178	fa	0.00008	hr	0.00001	-0.43495
mc00185.jpg	mc	0.99676	sl	0.00156	cb	0.00153	fa	0.00013	hr	0.00001	
sl00175_color_aug_bgr.jpg	sl	0.97021	cg	0.0261	ca	0.00245	hc	0.0011	cb	0.00011	0.00113
sl00175_color_aug_brg.jpg	sl	0.99201	cg	0.00709	hc	0.00041	cb	0.00024	ca	0.0002	0.02293
sl00175_color_aug_gbr.jpg	sl	0.9888	cg	0.01039	ca	0.00078	mc	0.00001	hc	0.00001	0.01972
sl00175_color_aug_gray.jpg	sl	0.51558	cg	0.46098	ca	0.01956	bv	0.00233	fa	0.0008	-0.4535
sl00175_color_aug_grb.jpg	sl	0.87983	cg	0.09826	ca	0.0217	cb	0.00012	mc	0.00006	-0.08925
sl00175_color_aug_rbg.jpg	sl	0.99	cg	0.00948	ca	0.00027	hc	0.00014	cb	0.00009	0.02092
sl00175_color_aug_rgb.jpg	sl	0.85709	cg	0.11941	ca	0.01336	hc	0.01	cj	0.00007	-0.11199
sl00175.jpg	sl	0.96908	cg	0.02708	ca	0.00267	hc	0.00104	cb	0.0001	0

14.4.4. Prediction with model retrained with dataset after 2nd augmentation

File Name	Top 1		Top 2		Top 3		Top 4		Top 5		Prediction score compare to origin photo
	Label	Score	Label	Score	Label	Score	Label	Score	Label	Score	
bv00011_color_aug_bgr.jpg	bv	0.99531	sl	0.00282	cg	0.00109	mc	0.0003	hc	0.00025	-6E-05
bv00011_color_aug_brg.jpg	bv	0.99962	cj	0.00035	mc	0.00001	sl	0.00001	fa	0.00001	0.00425
bv00011_color_aug_gbr.jpg	bv	0.98608	sl	0.01021	hc	0.00118	cg	0.00105	mc	0.00066	-0.00929
bv00011_color_aug_gray.jpg	bv	0.99111	cj	0.00262	sl	0.00249	mc	0.00152	cg	0.00083	-0.00426
bv00011_color_aug_grb.jpg	bv	0.99988	sl	0.00006	hc	0.00002	mc	0.00002	cg	0.00001	0.00451
bv00011_color_aug_rbg.jpg	bv	0.99987	cj	0.0001	hc	0.00001	sl	0	mc	0	0.0045
bv00011_color_aug_rgb.jpg	bv	0.99987	cj	0.00008	hc	0.00002	sl	0.00001	hr	0.00001	0.0045
bv00011.jpg	bv	0.99537	sl	0.0028	cg	0.00105	mc	0.00032	hc	0.00021	
ca00003_color_aug_bgr.jpg	ca	0.99913	cg	0.0006	fa	0.00023	sl	0.00003	cj	0.00002	0.00021
ca00003_color_aug_brg.jpg	ca	0.99761	fa	0.00104	cg	0.0008	cj	0.00048	sl	0.00007	-0.00131
ca00003_color_aug_gbr.jpg	ca	0.99774	cg	0.00141	sl	0.00078	fa	0.00005	cj	0.00002	-0.00118
ca00003_color_aug_gray.jpg	ca	0.99948	cg	0.00024	fa	0.00023	sl	0.00005	cj	0.00001	0.00056
ca00003_color_aug_grb.jpg	ca	0.99968	cg	0.0002	sl	0.00009	fa	0.00004	cj	0	0.00076
ca00003_color_aug_rbg.jpg	ca	0.99436	cg	0.00309	cj	0.00114	fa	0.0011	sl	0.0003	-0.00456
ca00003_color_aug_rgb.jpg	ca	0.99963	cg	0.00028	fa	0.00006	sl	0.00002	cj	0.00001	0.00071
ca00003.jpg	ca	0.99892	cg	0.0007	fa	0.00033	sl	0.00003	cj	0.00002	
cb00009_color_aug_bgr.jpg	cb	0.98665	sl	0.01323	cg	0.00005	mc	0.00004	fa	0.00003	0.00138
cb00009_color_aug_brg.jpg	cb	0.99928	sl	0.00067	fa	0.00004	cg	0.00001	mc	0	0.01401
cb00009_color_aug_gbr.jpg	cb	0.9817	sl	0.01773	cg	0.00043	fa	0.00008	mc	0.00006	-0.00357
cb00009_color_aug_gray.jpg	cb	0.51928	sl	0.44992	fa	0.02254	cg	0.00365	ca	0.00167	-0.46599
cb00009_color_aug_grb.jpg	cb	0.99239	sl	0.00705	cg	0.00028	fa	0.00026	mc	0.00001	0.00712
cb00009_color_aug_rbg.jpg	cb	0.98947	sl	0.00998	fa	0.00039	cg	0.00014	mc	0.00001	0.0042
cb00009_color_aug_rgb.jpg	cb	0.9771	sl	0.02225	fa	0.00038	cg	0.00023	mc	0.00001	-0.00817
cb00009.jpg	cb	0.98527	sl	0.01461	cg	0.00005	mc	0.00004	fa	0.00003	
cg00030_color_aug_bgr.jpg	cg	0.99967	hc	0.00033	sl	0	ca	0	fa	0	-0.00022
cg00030_color_aug_brg.jpg	cg	0.99746	hc	0.00251	sl	0.00002	fa	0.00001	ca	0	-0.00243
cg00030_color_aug_gbr.jpg	cg	0.99994	hc	0.00003	sl	0.00002	fa	0	ca	0	5E-05
cg00030_color_aug_gray.jpg	cg	0.99995	sl	0.00003	hc	0.00001	fa	0	ca	0	6E-05
cg00030_color_aug_grb.jpg	cg	0.99988	fa	0.00007	hc	0.00004	sl	0.00001	ca	0	-1E-05
cg00030_color_aug_rbg.jpg	cg	0.99945	hc	0.00052	sl	0.00002	fa	0.00001	ca	0	-0.00044
cg00030_color_aug_rgb.jpg	cg	0.99989	hc	0.0001	fa	0	ca	0	sl	0	0
cg00030.jpg	cg	0.99967	hc	0.00032	sl	0	ca	0	fa	0	
cj00010_color_aug_bgr.jpg	cj	0.99999	sl	0	fa	0	hc	0	cg	0	0
cj00010_color_aug_brg.jpg	cj	1	hc	0	sl	0	fa	0	cg	0	1E-05
cj00010_color_aug_gbr.jpg	cj	0.3728	cg	0.33201	sl	0.18155	ca	0.05882	fa	0.0504	-0.62719
cj00010_color_aug_gray.jpg	cj	0.99951	fa	0.00028	hc	0.00007	sl	0.00006	bv	0.00005	-0.00048
cj00010_color_aug_grb.jpg	cj	0.45199	cg	0.26157	sl	0.14093	ca	0.06605	hc	0.04677	-0.548
cj00010_color_aug_rbg.jpg	cj	0.99998	hc	0.00001	fa	0	sl	0	cg	0	-1E-05
cj00010_color_aug_rgb.jpg	cj	0.90654	cg	0.02887	sl	0.02784	fa	0.01565	ca	0.01134	-0.09345
cj00010.jpg	cj	0.99999	sl	0	fa	0	hc	0	cg	0	
fa00014_color_aug_bgr.jpg	fa	0.99884	ca	0.00054	bv	0.00036	cg	0.00015	sl	0.00011	-5E-05
fa00014_color_aug_brg.jpg	fa	0.99848	ca	0.00069	bv	0.00048	sl	0.00018	cg	0.00016	-0.00041
fa00014_color_aug_gbr.jpg	fa	0.96733	ca	0.02412	sl	0.00398	cg	0.00381	bv	0.00069	-0.03156
fa00014_color_aug_gray.jpg	fa	0.96368	sl	0.0272	ca	0.00761	bv	0.00089	cg	0.00054	-0.03521
fa00014_color_aug_grb.jpg	fa	0.91318	ca	0.05972	sl	0.01883	cg	0.00704	bv	0.00085	-0.08571
fa00014_color_aug_rbg.jpg	fa	0.99838	bv	0.00132	sl	0.00018	ca	0.00009	cg	0.00001	-0.00051
fa00014_color_aug_rgb.jpg	fa	0.84987	ca	0.09702	sl	0.03124	cg	0.01148	bv	0.00983	-0.14902

fa00014.jpg	fa	0.99889	ca	0.00051	bv	0.00034	cg	0.00014	sl	0.00012	
hc00033_color_aug_bgr.jpg	hc	0.99829	cb	0.00164	cj	0.00005	bv	0.00001	hr	0.00001	-3E-05
hc00033_color_aug_brg.jpg	hc	0.99546	cb	0.00437	bv	0.00009	cj	0.00004	hr	0.00003	-0.00286
hc00033_color_aug_gbr.jpg	hc	0.58121	cb	0.41794	sl	0.00057	cg	0.00012	bv	0.00012	-0.41711
hc00033_color_aug_gray.jpg	hc	0.5986	cb	0.39267	bv	0.00838	sl	0.00025	cg	0.00004	-0.39972
hc00033_color_aug_grb.jpg	hc	0.75929	cb	0.24023	bv	0.0003	sl	0.00013	cg	0.00004	-0.23903
hc00033_color_aug_rbg.jpg	hc	0.99649	cb	0.00286	bv	0.00047	hr	0.00008	cj	0.00007	-0.00183
hc00033_color_aug_rgb.jpg	hc	0.99987	cb	0.0001	bv	0.00002	cg	0	hr	0	0.00155
hc00033.jpg	hc	0.99832	cb	0.0016	cj	0.00005	hr	0.00001	bv	0.00001	
hr00035_color_aug_bgr.jpg	hr	0.99994	hc	0.00006	bv	0	cj	0	sl	0	0
hr00035_color_aug_brg.jpg	hr	0.99999	hc	0.00001	cj	0	bv	0	sl	0	5E-05
hr00035_color_aug_gbr.jpg	hr	0.99501	hc	0.00378	bv	0.00064	sl	0.00056	cb	0.00001	-0.00493
hr00035_color_aug_gray.jpg	hr	0.99873	bv	0.00086	hc	0.0004	cj	0.00001	sl	0	-0.00121
hr00035_color_aug_grb.jpg	hr	0.99693	hc	0.00171	sl	0.00087	bv	0.00048	fa	0.00001	-0.00301
hr00035_color_aug_rbg.jpg	hr	0.99998	hc	0.00001	bv	0	cj	0	sl	0	4E-05
hr00035_color_aug_rgb.jpg	hr	0.99892	hc	0.00087	bv	0.00017	sl	0.00004	fa	0	-0.00102
hr00035.jpg	hr	0.99994	hc	0.00005	bv	0	cj	0	sl	0	
mc00185_color_aug_bgr.jpg	mc	0.99652	sl	0.00303	cb	0.00033	fa	0.00009	hr	0.00002	-0.00042
mc00185_color_aug_brg.jpg	mc	0.99303	sl	0.00665	cb	0.00024	hr	0.00004	fa	0.00004	-0.00391
mc00185_color_aug_gbr.jpg	mc	0.96894	sl	0.02888	cb	0.00159	fa	0.00055	bv	0.00001	-0.028
mc00185_color_aug_gray.jpg	cb	0.67774	mc	0.16399	sl	0.15066	fa	0.007	hr	0.00028	-0.83295
mc00185_color_aug_grb.jpg	mc	0.75333	sl	0.21125	cb	0.03522	fa	0.0001	cg	0.00003	-0.24361
mc00185_color_aug_rbg.jpg	mc	0.99626	sl	0.00361	cb	0.00012	fa	0.00001	hr	0	-0.00068
mc00185_color_aug_rgb.jpg	mc	0.56803	sl	0.30536	cb	0.12644	hr	0.00005	fa	0.00004	-0.42891
mc00185.jpg	mc	0.99694	sl	0.00266	cb	0.00029	fa	0.00009	hr	0.00002	
sl00175_color_aug_bgr.jpg	sl	0.94103	cg	0.04894	ca	0.00896	hc	0.00086	cb	0.00016	0.00316
sl00175_color_aug_brg.jpg	sl	0.98152	cg	0.01661	ca	0.00089	hc	0.0005	cb	0.00035	0.04365
sl00175_color_aug_gbr.jpg	sl	0.95977	cg	0.03802	ca	0.00215	hc	0.00002	cb	0.00002	0.0219
sl00175_color_aug_gray.jpg	cg	0.79955	sl	0.16578	ca	0.03231	fa	0.00052	hc	0.0005	-0.77209
sl00175_color_aug_grb.jpg	sl	0.50448	cg	0.44594	ca	0.049	cb	0.00036	hc	0.00014	-0.43339
sl00175_color_aug_rbg.jpg	sl	0.97659	cg	0.02236	ca	0.00082	hc	0.00014	cb	0.00006	0.03872
sl00175_color_aug_rgb.jpg	sl	0.67346	cg	0.27952	ca	0.03638	hc	0.01042	cj	0.00016	-0.26441
sl00175.jpg	sl	0.93787	cg	0.05153	ca	0.00959	hc	0.00082	cb	0.00016	0

14.4.5. Summary of average improvement

Method	**Average Improvement**		
	Model retrained with origin dataset	**Model retrained with dataset after 1st augmentation**	**Model retrained with dataset after 2nd augmentation**
bgr	1.2E-05	0.000148	0.000397
brg	0.000704	0.003805	0.005105
gbr	-0.136841	-0.171108	-0.110088
gray	-0.126833	-0.290352	-0.251129
grb	-0.12409	-0.194328	-0.154037
rbg	-0.000545	0.00342	0.003943
MAX	0.000704	0.003805	0.005105
MIN	-0.136841	-0.290352	-0.251129

14.5. Visualization of learned features regarding color-changed images

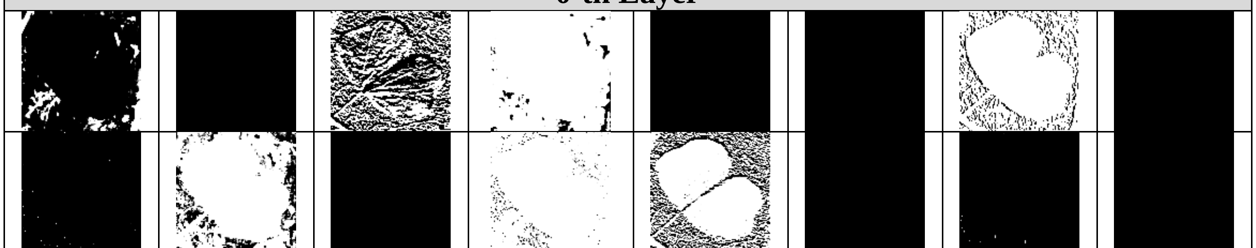
14.5.1. Learned features of images bv00011_color_aug_bgr

Input Image

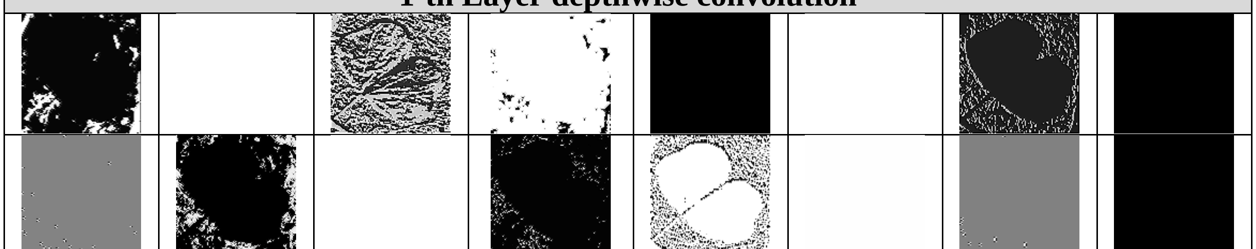
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

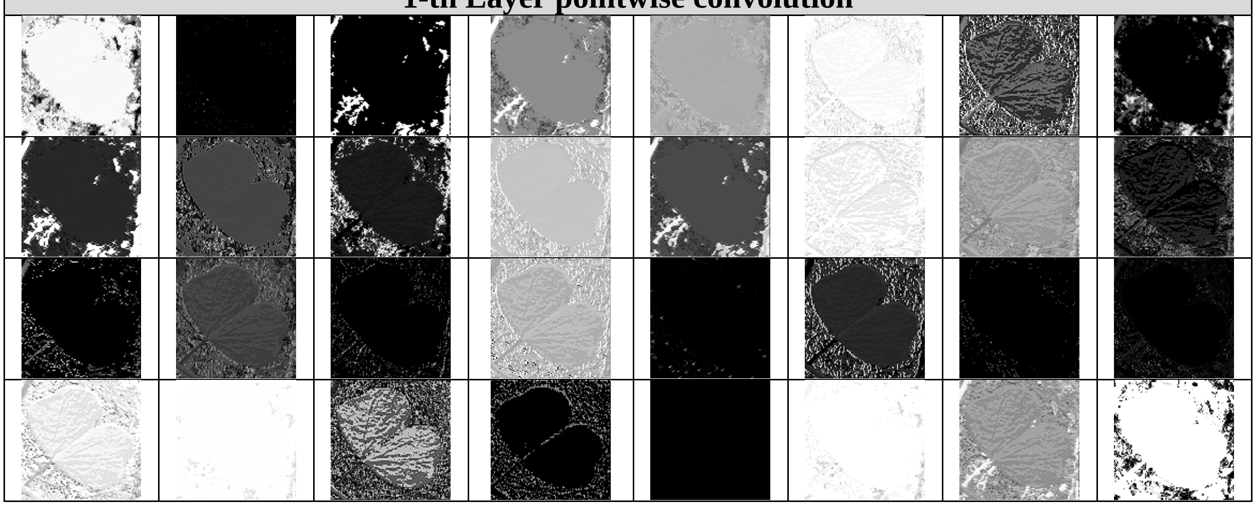
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



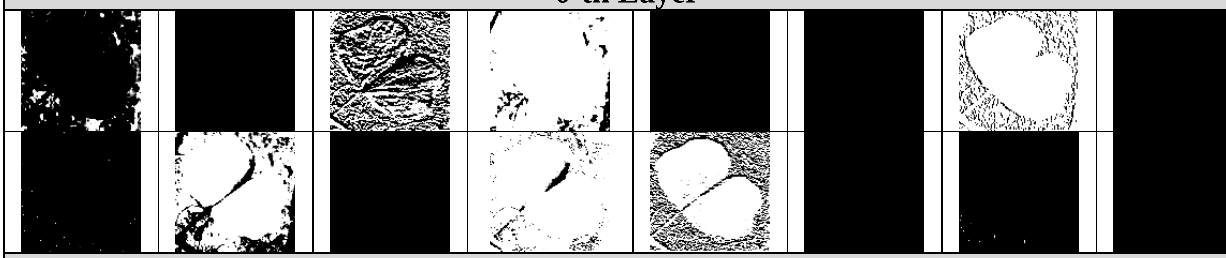
14.5.2. Learned features of images bv00011_color_aug_gbr

Input Image

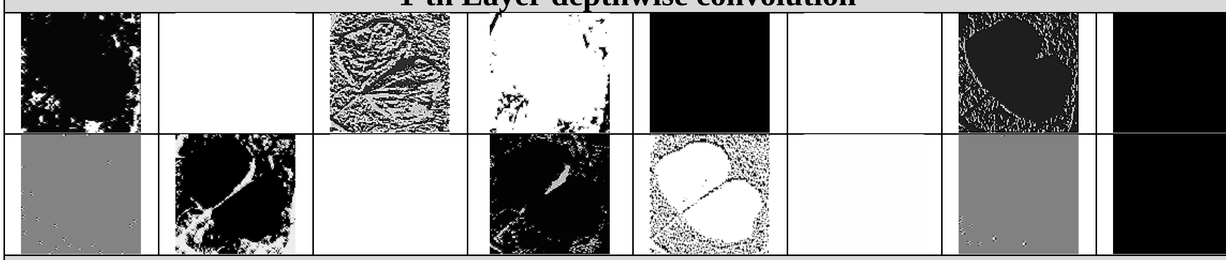
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

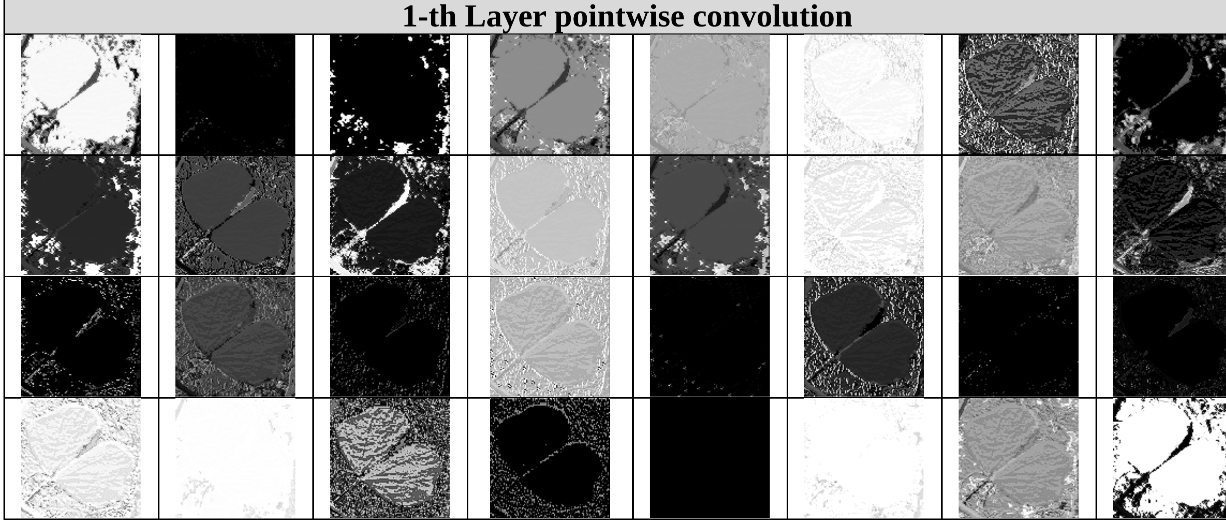
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



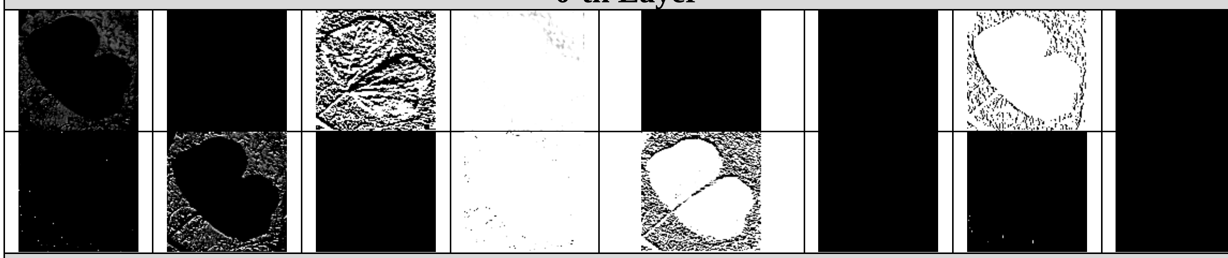
14.5.3. Learned features of images bv00011_color_aug_gray

Input Image

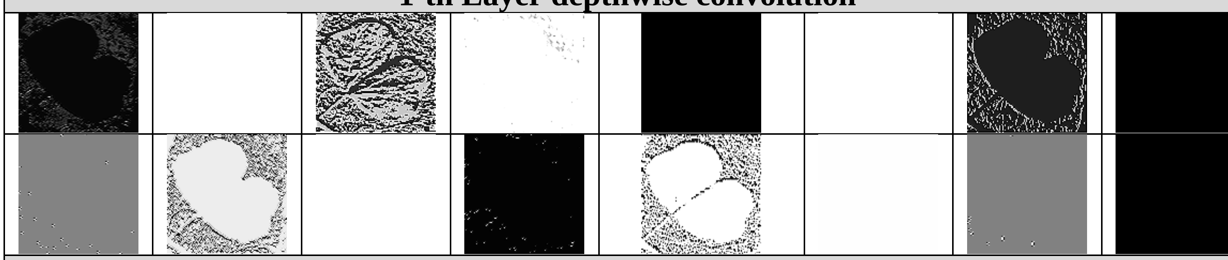
Index of output in a layer:

0-th Layer							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
1-th Layer Depthwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
1-th Layer pointwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

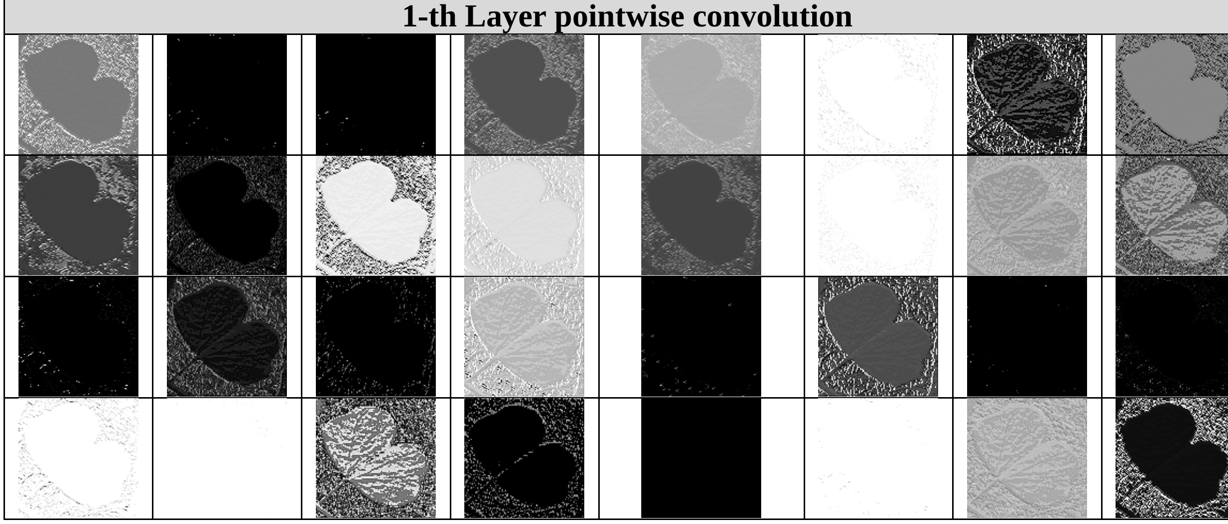
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



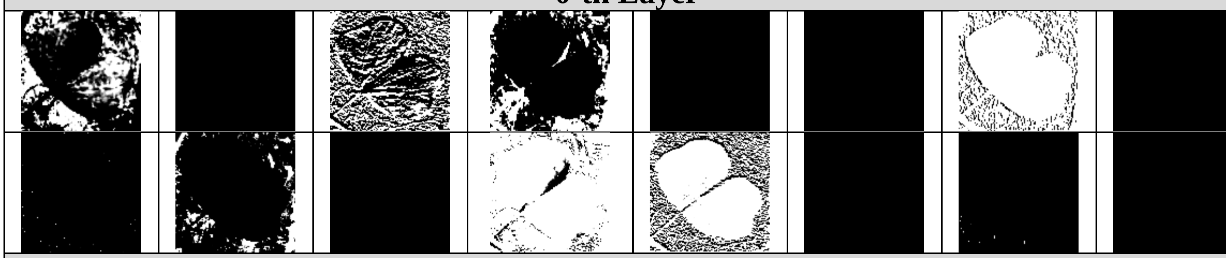
14.5.4. Learned features of images bv00011_color_aug_grb

Input Image

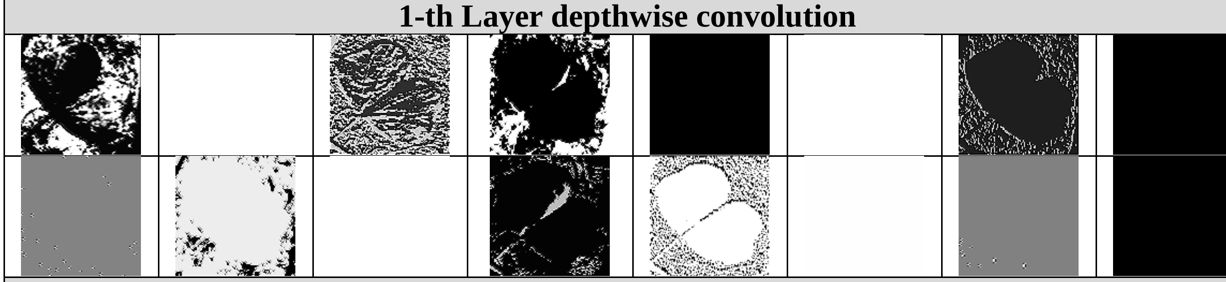
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

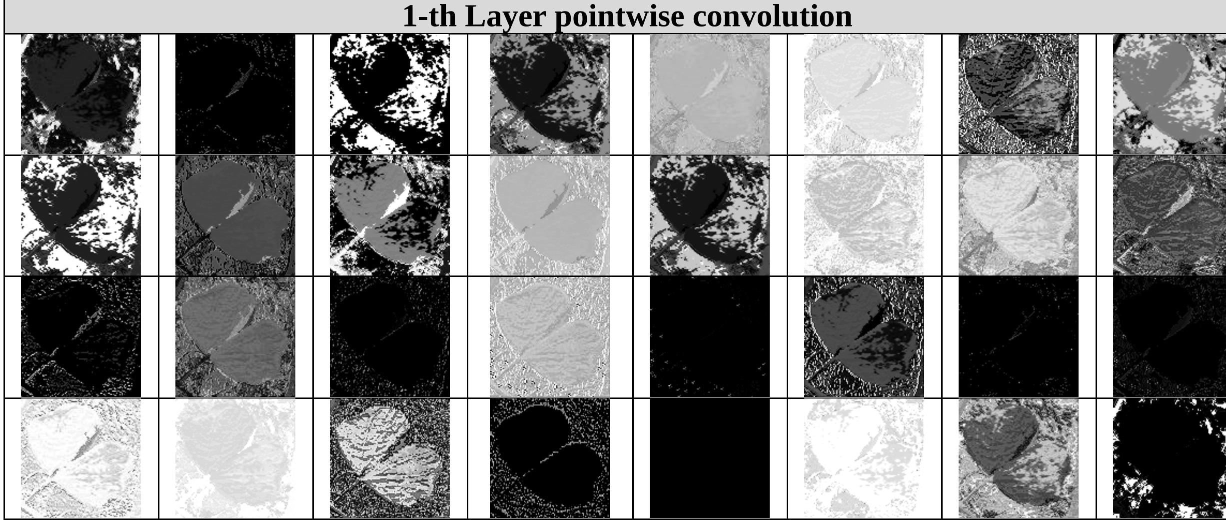
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



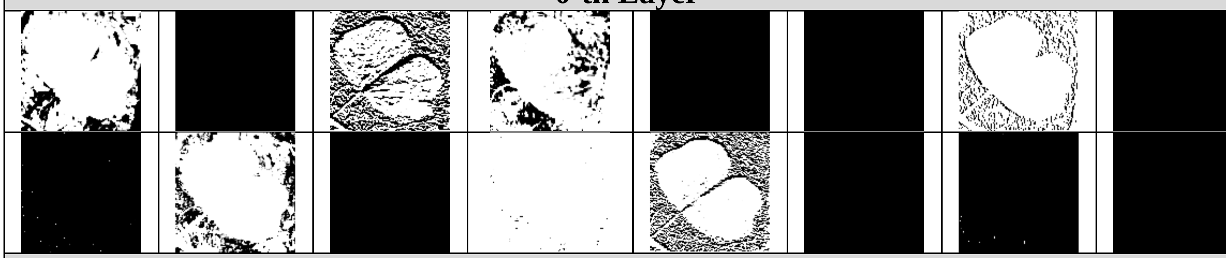
14.5.5. Learned features of images bv00011_color_aug_rgb

Input Image

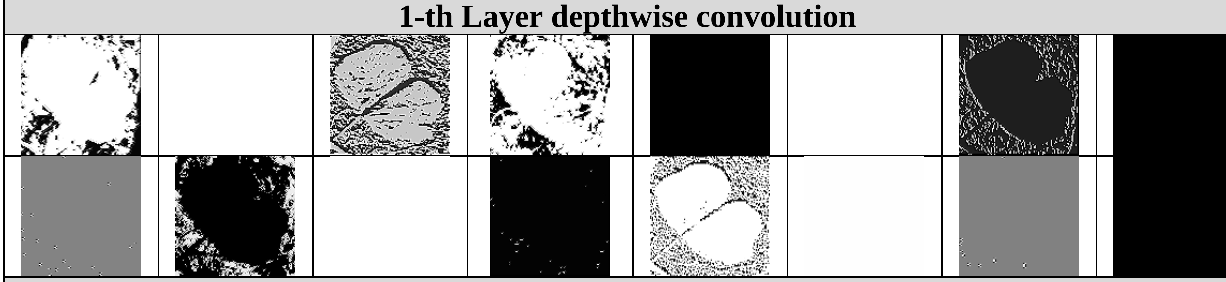
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

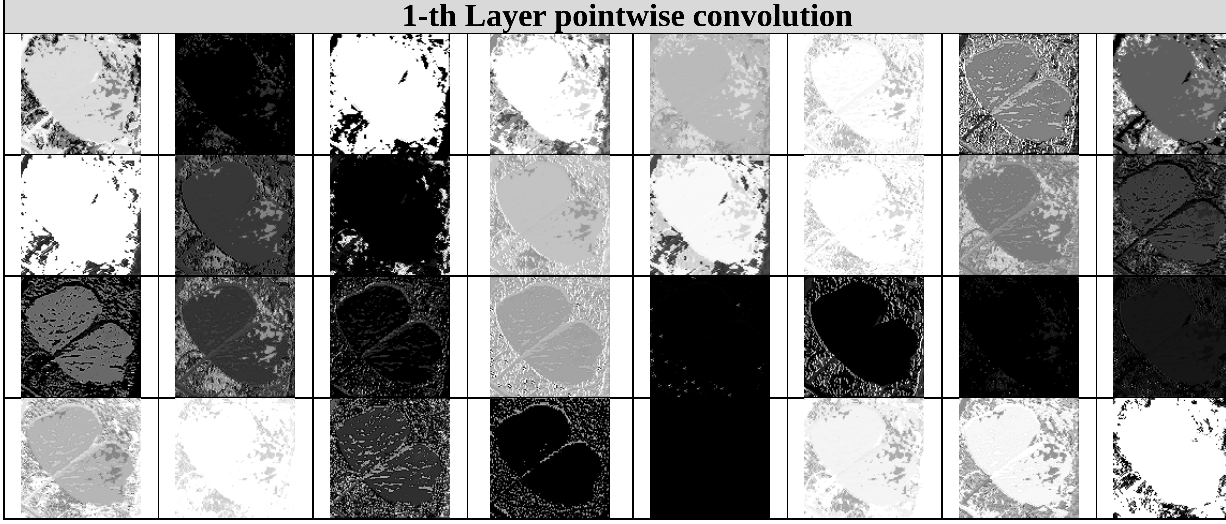
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



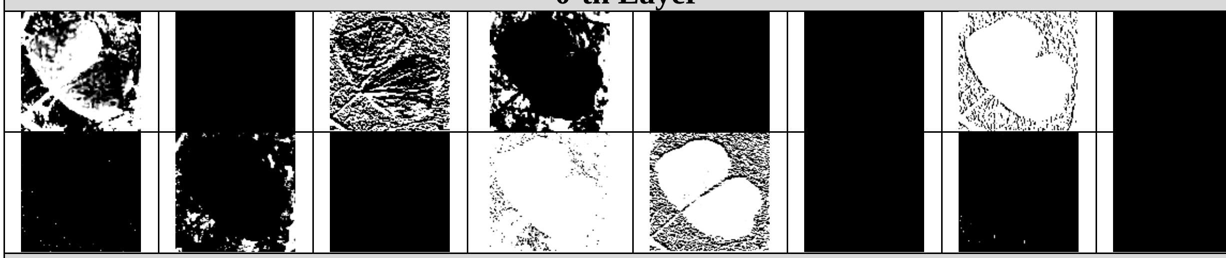
14.5.6. Learned features of images bv00011_color_aug_brg

Input Image

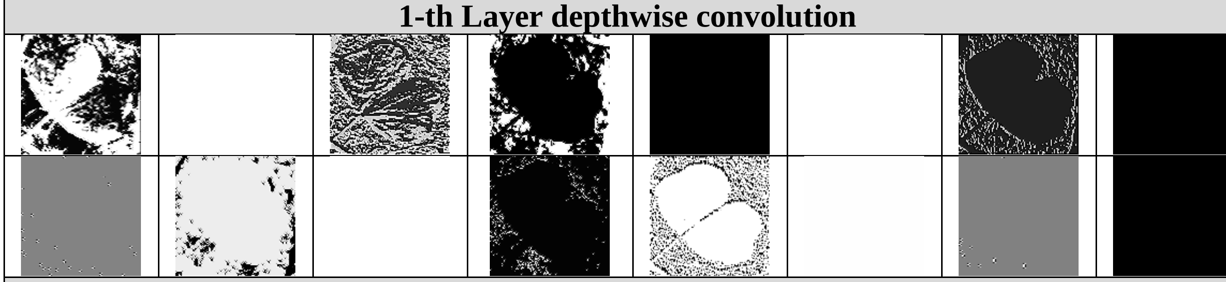
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

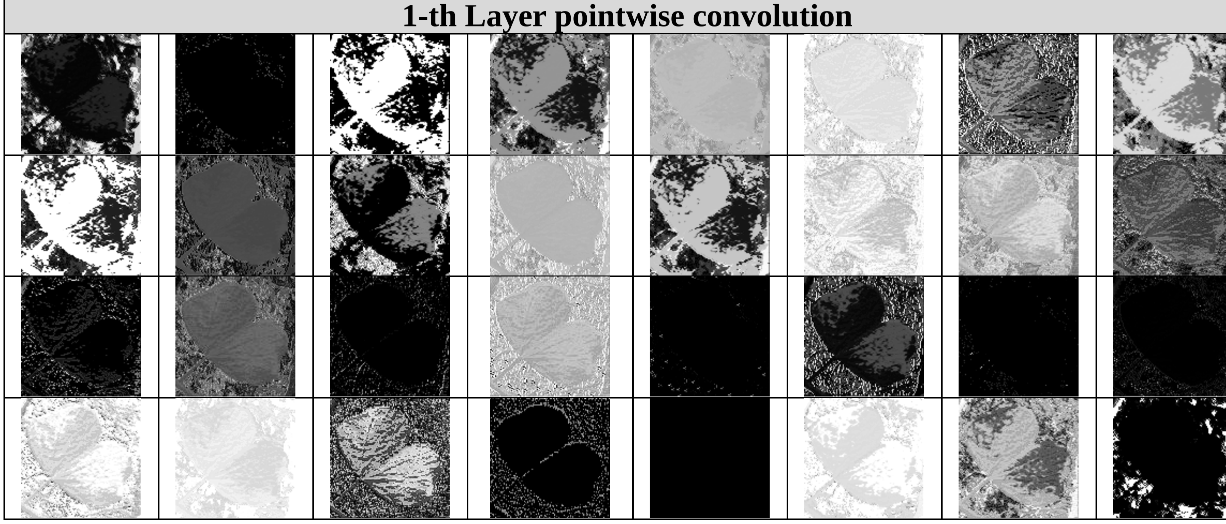
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



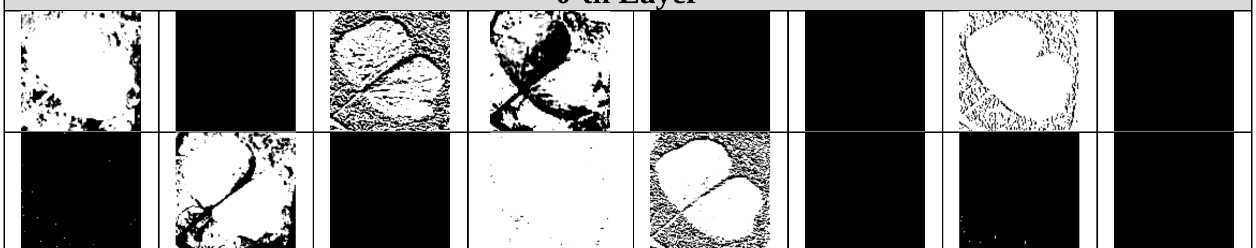
14.5.7. Learned features of images bv00011_color_aug_rbg

Input Image

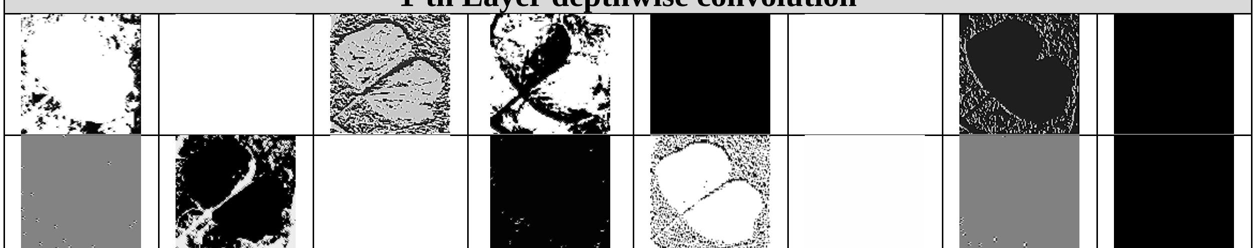
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

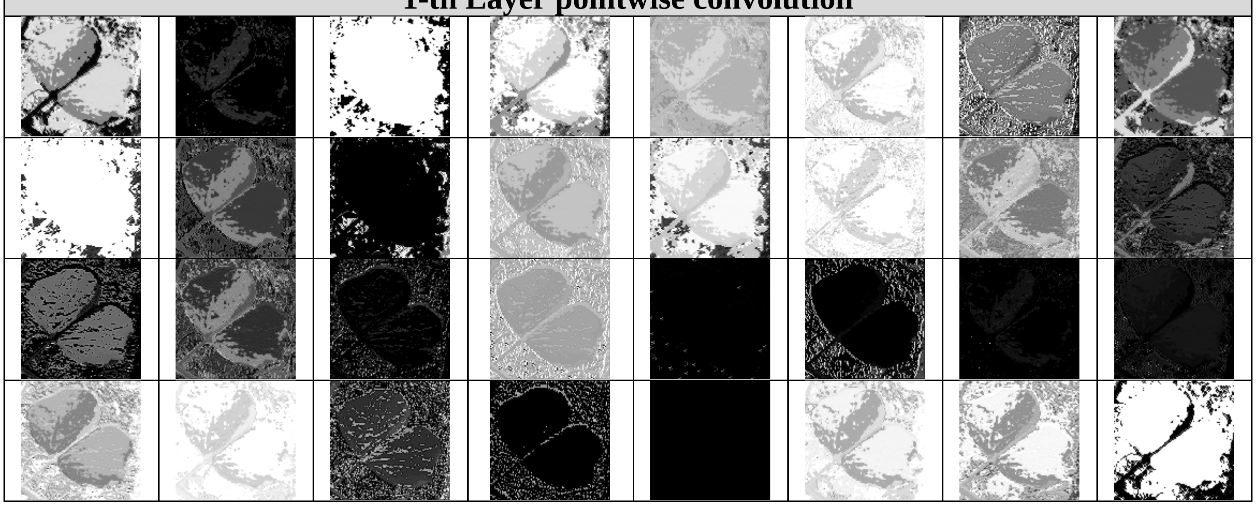
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



14.6. Visualization of learned features regarding dried leaves

14.6.1. Learned features of image bv00011_dried_1

Input Image

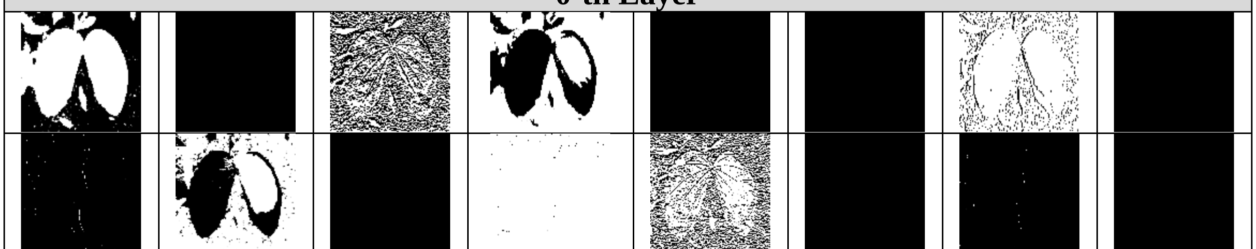
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*

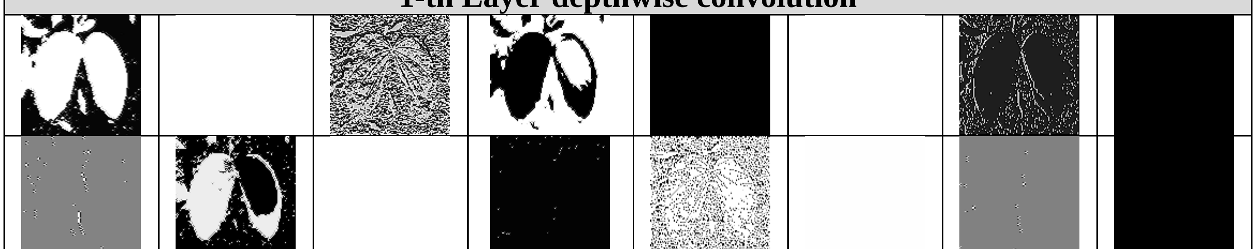
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*

1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

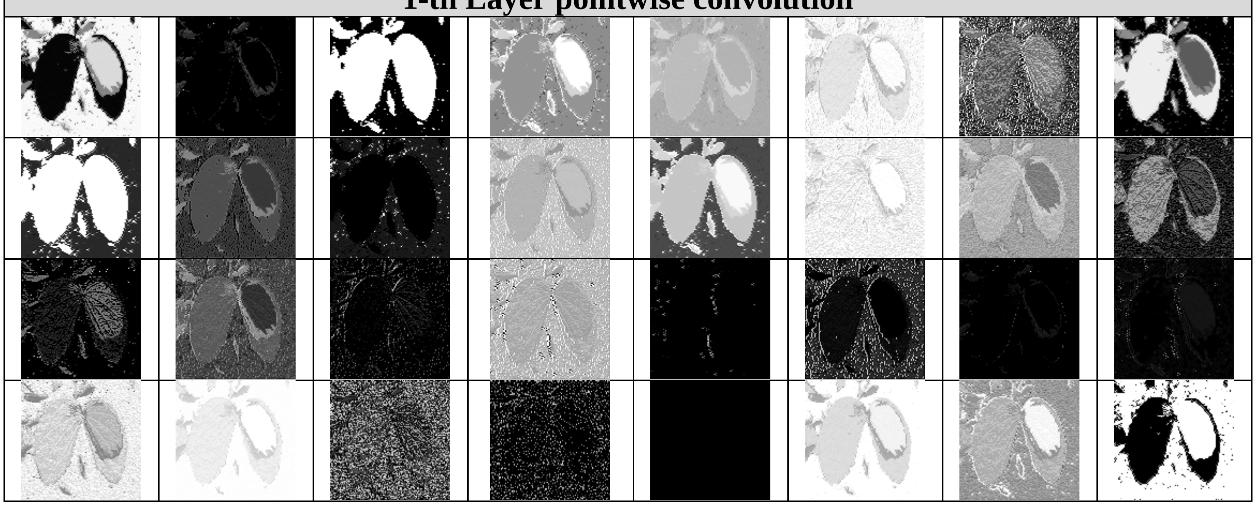
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



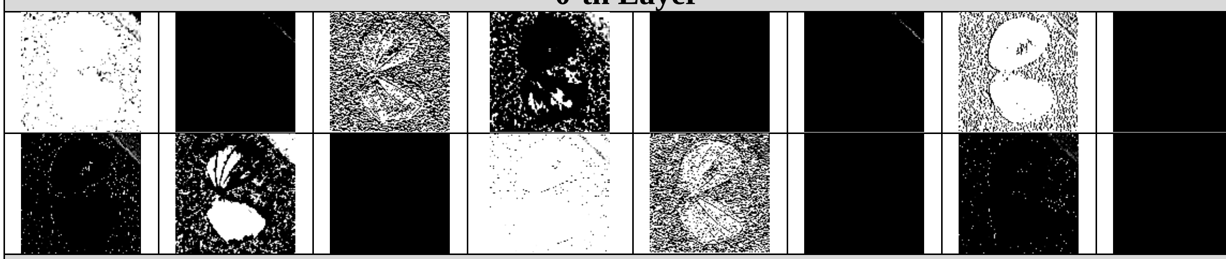
14.6.2. Learned features of image bv00011_dried_2

Input Image

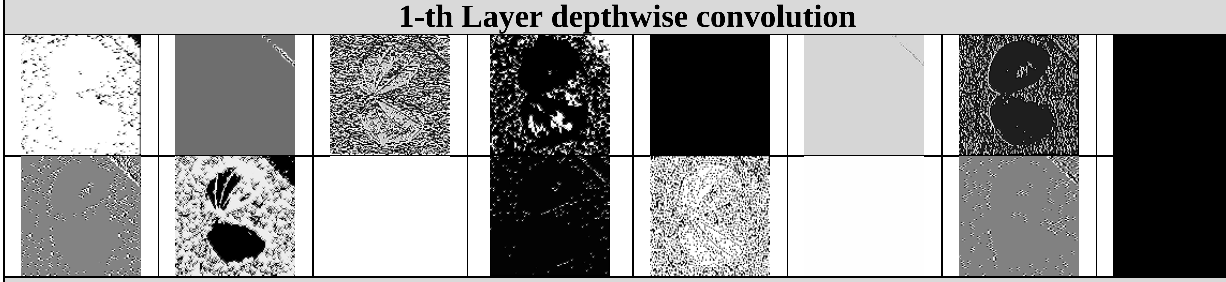
Index of output in a layer:

0-th Layer							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer Depthwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1-th Layer pointwise convolution							
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
9	*10*	*11*	*12*	*13*	*14*	*15*	*16*
17	*18*	*19*	*20*	*21*	*22*	*23*	*24*
25	*26*	*27*	*28*	*29*	*30*	*31*	*32*

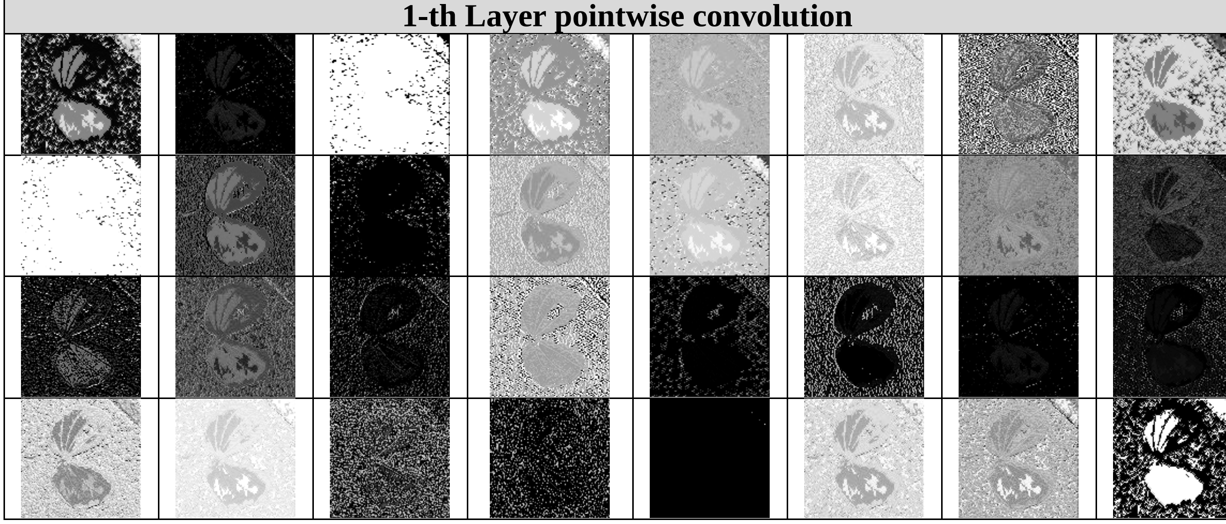
0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution



14.6.3. Learned features of image bv00011_dried_3

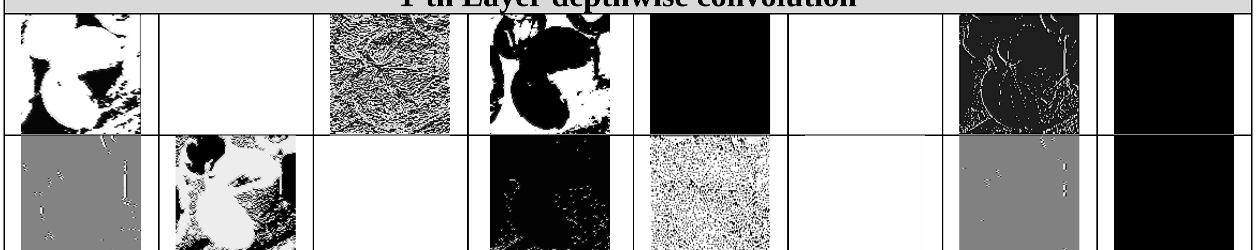
Input Image

Index of output in a layer:

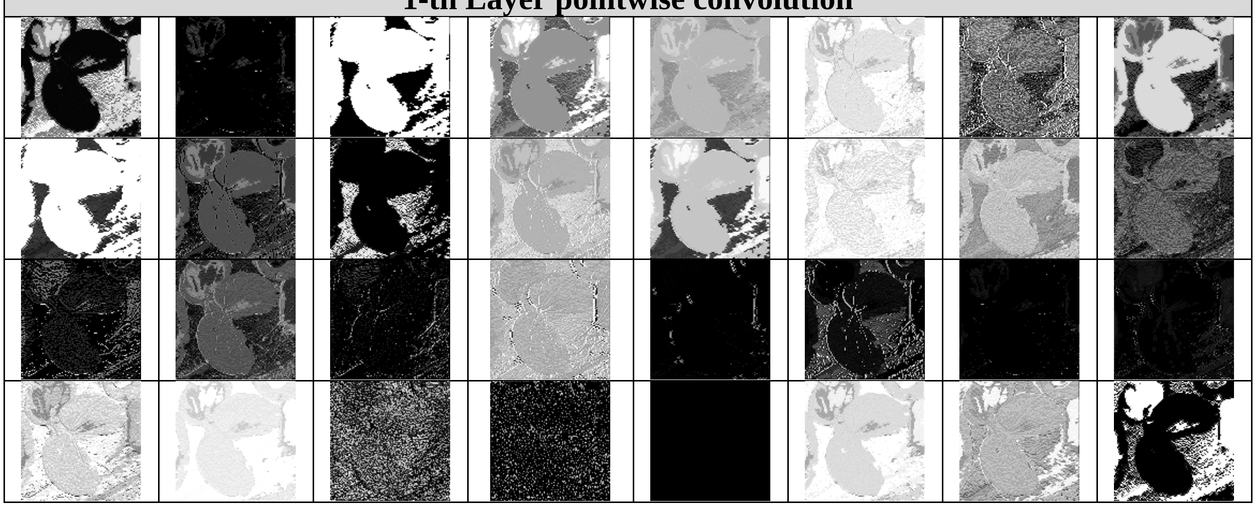
0-th Layer							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
1-th Layer Depthwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
1-th Layer pointwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

0-th Layer

1-th Layer depthwise convolution



1-th Layer pointwise convolution



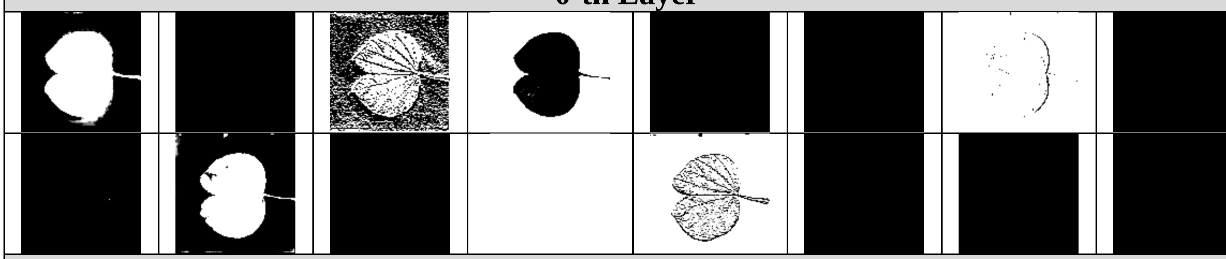
14.6.4. Learned features of image bv00011_dried_4

Input Image

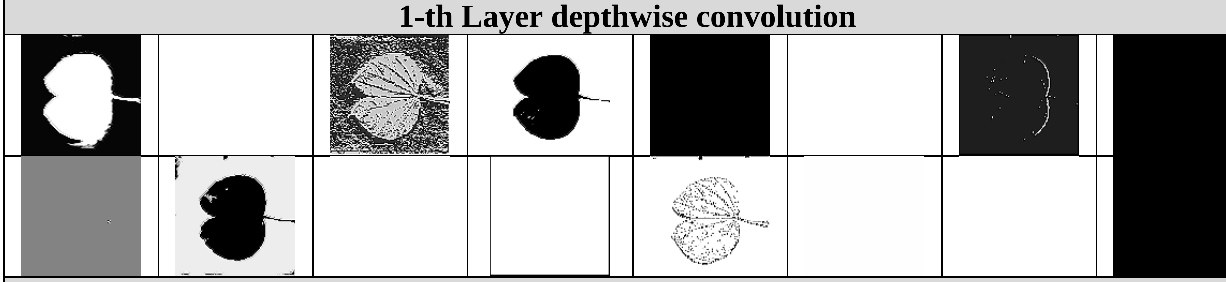
Index of output in a layer:

0-th Layer							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
1-th Layer Depthwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
1-th Layer pointwise convolution							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

0-th Layer



1-th Layer depthwise convolution



1-th Layer pointwise convolution