THE UNIVERSITY OF MELBOURNE

An IOS Application for Plant Image Recognition using Machine Learning Approaches

COMP90055
Computing Project (25 Credits)

Name: Min Gao
Student Number:773090

Name: Lang Lin
Student Number:772341

Type of Project: Software Development Project
Supervisor: Prof. Richard Sinnott
Semester 1, 2017

Acknowledgement

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude We give to our computing project supervisor, Prof. Richard Sinnott, whose contribution in suggestions and encouragement, helped us to coordinate our project.

Abstract

Vision is an imperative perception for the human to know the world. It is a very simple task for the human to visually identify objects in a picture or to find objects in a photo. However, let the computer to recognise the contents of the image is not an easy job. Not to mention let the computer to recognise flowers which are similar to each other. The purpose of image recognition problem is using software to process, analyze and understand the contents of the image, so that the computer can automatically identify a variety of different models of targets and objects from the image. The method of deep learning is the breakthrough of this problem. So that the identification ability of software can close to the level of manual annotation.

In addition, there are many kinds of flowers on the earth. Ordinary people have limited knowledge about the flowers. Using deep learning method to implement an IOS application for flower recognition can help people know the world better.

In this software development project, massive data of flower image were crawled from the web at first. Preprocessing for these images were done to get two relatively pure data set (50-classification and 125-classification). Then these data set were trained through the convolutional neural network to get the classifier model (50-classifier and 125-classifier). And finally, an IOS application for flower image recognition called knowFlower was implemented to help people identify different flowers.

key words: Deep Learning, Tensorflow, CNN, Computer Vision, Image Procressing, Data Mining, IOS Application, Flower

We certify that

- *this thesis does not incorporate without acknowledgment any material previously*
- *submitted for a degree or diploma in any university; and that to be best of our knowledge and belief it does not contain any material previously published or written by another person where due reference is not made in the test.*
- *where necessary we have received clearance for this research from the University's Ethics Committee and have submitted all required data the Department*
- *the thesis is 6200 words in length (excluding text in images, table, bibliographies and appendices).*

Contents

Figures

1 Introduction

1.1 Mobile Application

Smartphones are becoming increasingly popular around people these years. According to data, there are 2.32 billion people using smartphones all over the world in 2017, and the trend of the number of individuals using the smartphone is still rising. In prediction, there will 2.87 billion using smartphones in 2020. Applications on smartphones help users to pay the bill, learn knowledge, browse news, play games, watch the video, etc. With no doubt that smartphones are indispensable for modern people now and applications are smartphones' soul.

Convenient applications help smartphone users to get rid of watching weather forecast on TV to know the intraday weather, checking papery dictionary to master the meaning of a phrase, monitor store surveillance video anywhere instead of computer only. It will help people more if there is an application which is capable of recognizing a flower by a photo.

The first aim of the application is to enable ordinary people can easily know a flower's species simply by taking a picture of this flower and using the application to get a result.

The second aim of the application is to help the student gain incognizant knowledge easier, for example when students find an unknown flower on practice course they can simply tap their fingers on the smartphone to recognize flower species rather than search for this information after class.

The goal of this project is to create an IOS application about plant image recognition just like all other applications which can help people's daily life easier.

To achieve this goal, the project used plenty of techniques like deep learning, image processing, IOS programming.

1.2 Deep Learning

With the alphaGo beating Li Shishi, deep learning has become a very hot topic. Behind the latest technology such as real-time translation and unmanned driving, deep learning is the core force to promote the development of these technologies.

In the past, traditional machine learning method was used to solve some problems. For many machine learning problems, feature extraction is not a simple task. For some complex challenges, it takes a lot of time and effort to design effective feature sets through manual methods. Since manual methods are not good for extracting the features of an entity, is these an automatic way of doing that? The answer is yes. One of the key problems the deep learning has solved is that it can automatically combine simple feature into more complex features, and use these combinatorial features to solve problems. Deep learning is a branch of machine learning (Figure 1). In addition to learning the association between features and tasks, it can automatically extract more complex features from simple features.

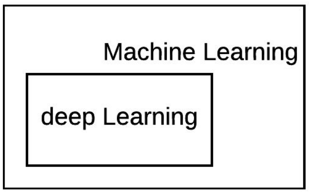

Figure 1 relationship between machine learning and deep learning

Figure 2 shows the difference in process between the traditional machine learning and deep learning. The deep learning algorithm can learn more complex feature from data, which makes the last step weight learning easier and more effective.

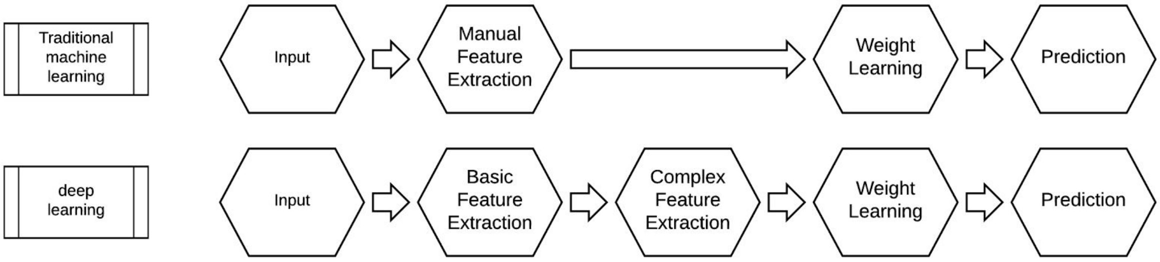

Figure 2 difference between traditional machine learning and deep learning

However, the traditional deep learning architecture fully connected neural network can not process image data well because the biggest problem with this structure is that there are too many parameters in the full connection layer especially when dealing with images. Too many parameters can lead to slow calculation and overfitting problem. It is impossible to train such a neural network on the regular computer and even the model trained by lots of time will not work very well. Therefore, a more logical neural network structure is needed to reduce the number of parameters in the neural network effectively and to train more reliable models， convolutional neural network (CNN) was used in this project. Figure 3 shows the structural difference between the fully connected neural network and CNN.

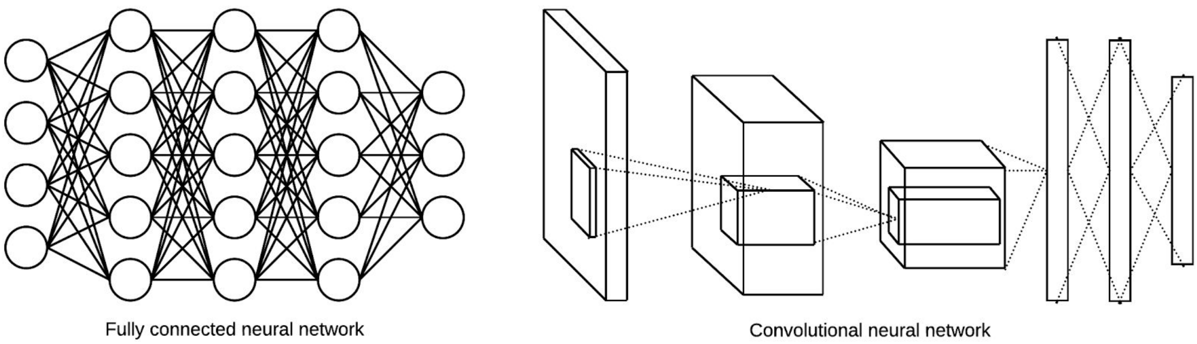

Figure 3 difference between fully connected neural network and CNN

By using convolution layer, regardless of the size of the image, the number of weights we need to train depends only on the size of the convolution kernel and the number of convolution kernels. Then we can process images of any size with fewer parameters. At the same time, the training efficiency is significantly improved, and the ability to express features is much stronger.

1.3 Tensorflow

To apply deep learning to the practical problem, a deep learning tool needs to be chosen. Tensorflow is an open source software framework for numerical computation using data flow graphs. It can support a variety of deep learning algorithms. In this project, Tensorflow was used to build the convolutional neural network and train the dataset.

2 Dataset

Dataset is an indispensable portion of Deep Learning, strong algorithm and pure dataset are most important keys for deep learning to train a model with great classification ability. Therefore, build a pure dataset is the first job in this project.

2.1 Crawling Dataset

Since no pre-prepared dataset exists, all data being used in this project is downloaded from Google Image and Flickr by their public API through the Internet. Two different datasets are downloaded for case studies: one contains 125 flowers differentiate by species according to Burnley Plant Guide's "Bulbs, Corms, Rhizomes" category[5]; another contains 50 flowers differentiate by genus according to 50 most beautiful flowers in the world[6]. The reason of making two different case studies is that in the 50 flower genera case, most flower's appearance can easily be differentiated by shape or color or other factors because even two flower genera under the same family they usually have different appearance like it showed in figure 4. 125 flower species case's flower comes from Burnley Plant Guide, the advantage of it is the website has some extra information like plant normal height and weight range at maturity and environment tolerance knowledge which can be used to help classify flowers better in future work. The disadvantage is that flowers belong to the same category Burnley Plant Guide defined usually are flower species under the same genus, like Bulbine bulosa and Bulbine semibarbata in 125 flower species case which are showed in figure 5. It is difficult to identify a flower belongs to Bulbine bulosa or Bulbine semibarbata because their appearance are mostly the same and if not enough images contains their differentia -- stamen then there won't be a big weight for stamen while training the 125 flower species classification model result in the model unable to differentiate Bulbine bulosa and Bulbine semibarbata.

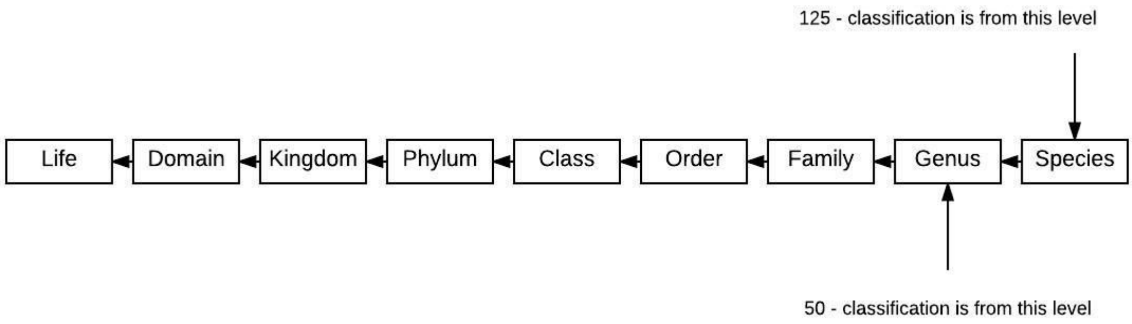

Figure 4 Taxonomic rank

Figure 5 Bulbine bulosa(left) and Bulbine semibarbata(right)

This project crawls 2000 images from Flickr, 1000 images from Google Image each genus set for 50 flower genera classification case and 1000 images from Flickr, 500 images from Google Image each species set for 125 flower species classification case. All images are crawled while images' text, tag or description contains the flower name which is used as the keyword to search. So if an image doesn't contain any flower in it but its text, tag or description contains flower's name then it will be downloaded. For example, an image has a tag like "Rose Tattoo" or has a description like "Jack and Rose's story in Titanic is so beautiful" will be downloaded into "Rose" set.

While downloading the image, some flowers' names are the same as other famous things like "stock", "jasmine", "violet", and so on. To prevent the image sets contain too many disturbance terms, these kinds of flowers' names are followed by an extra text "flower" while being searched and crawled on the Internet.

2.2 Purify Dataset

Machine learning requires a pure dataset to build a model with great ability of prediction after training, testing and validating. Usually, for a good algorithm purer the dataset it uses, better the classify result it will be. As a branch of machine learning, deep learning inherits this property from machine learning. So, for the purpose of gaining a fair result in this project, next work to do is to purify dataset downloaded from Google Image and Flickr. For this project's datasets, purification is separated into two procedures: first procedure is to purify dataset by using deep learning and Tensorflow to remove all images in dataset which did not contain any flower; second procedure still uses deep learning and Tensorflow, it delete all images which did not contain correct flower in all flower sets.

2.2.1 Procedure One

In the first procedure, 3000 images do not contain flower are downloaded into a set named "No_Flower", then from each genus/species 50 images were selected and copied into a set named "Yes_Flower". Images did not contain flower in "Yes_Flower" set need to be removed manually because when deep learning building model according to features learned from "No_Flower" and "Yes_Flower" sets, "No_Flower" set contains various kinds of pictures which make deep learning cannot learn a fixed feature structure while "Yes_Flower" set contains only flower images can let deep learning easily learn flower features' structure to let the model being built with a great ability of identify whether an image contains flower or not by simply classify the image into "Yes_Flower" set or "No_Flower" set. There is a small trick for two different cases: "No_Flower" set can use the same set while "Yes_Flower" need different sets because there is some kinds of flower have very special appearance itself like figure 6 (from 50 flower genera case study) and figure 7 (from 125 flower species case study), deep learning need to learn their features by adding these flowers' images into "Yes_Flower" set otherwise while they will be classified into "No_Flower" set because "Yes_Flower" set's features are fewer but better, "No_Flower" set's features are many but complex so any feature not seen in "Yes_Flower" set will be considered as part of "No_Flower" set.

Figure 6 Bird of paradise

Figure 7 Albuca bracteata

2.2.2 Procedure Two

The second procedure also needs two different case studies have their own dataset. In this procedure, each flower genus/species selects 50 images and copy them into "Sample" set, then manually remove any image whose content is not the same as its flower name. Then one flower genus/species in a time from "Sample" set copy all images belong to this genus/species into "Right_Kind" set and others into "Wrong_Kind" set. Total number of images in "Right_Kind" set is count, then a model to classify "Right_Kind" set and "Wrong_Kind" set was trained by deep learning, this model is being used to classify whether images in the original flower set have correct content or not and if this model judge an image belongs to "Right_Kind" set then copy this image into "Right_Kind" set. After these processes the total number of images in "Right_Kind" set is count again, if the number of images in "Right_Kind" set before and after testify image content are not the same, then redo previous processes until the number of images are the same. Finally, using the final model to judge whether images in the original flower set belongs to "Right_Kind" set or "Wrong_Kind" set. If there is more than 70% chance that this image belongs to "Wrong_Kind" set then this image will be removed from the original flower set. The reason of using 70% is because there are many flowers looks like each other to affect the classified result, if only use image belong to "Right_Kind" set it might cause after purification there are too little images left to train a good model.

After purification, each genus/species set in both case studies only left around 500 images or less. There might still have little wrong images because these images are classified into "Yes_Flower" set and "Right_Kind" set during classification procedures. But since deep learning is learning image features automatically so these wrong images are acceptable because they won't affect the result of training too much and there is no better way to build a 100% pure dataset right now.

3 Image Preprocessing

In order to improve the accuracy of image recognition and the training efficiency, it is very important to preprocess the image of the dataset. As we all know, luminance, contrast ratio, and some other attributes have a great impact on the image. The same object varies greatly when either contrast ratio or luminance changes in circumstances. However, these should not affect the final recognition result. Therefore, image flip and image attributes adjustment should be done to make sure that the trained neural network model can identify flowers in different conditions. In addition, the image crawled from the web is not fixed, but the number of input nodes of the neural network is fixed. Therefore, the size of the image needs to be unified before putting into the neural network, so we also need to adjust the size of the image.

3.1 Image Resizing

There are generally three algorithms for resizing images: bilinear interpolation, nearest neighbor interpolation and bicubic interpolation. Resizing image with nearest neighbor interpolation do not have a satisfactory result. While the calculation of bicubic interpolation is too large. Therefore, in this project, bilinear interpolation is used to adjust the size of the image. The core of this algorithm is to determine the pixel value of the target coordinates by the pixel values of the four coordinate around the target coordinates in the original image. Following formulas shows how it works:

$$f(x+u, y+v) = (1-u)(1-v)f(x,y) + (1-u)vf(x,y+1) + u(1-v)f(x+1,y) + uv(x+1,y+1)$$

(x, y are the integer part of the floating-point coordinate. u, v are the decimal part of the floating-point coordinate)

Figure 8 shows 322x359 pixels sized image adjusted to 299x299 pixels sized image. As you can see from the image, the features of the image after resizing are still obvious. By resizing the image, all size of images are unified. Besides, especially resize the large image into the small image, the efficiency of the training of the CNN will be greatly improved.

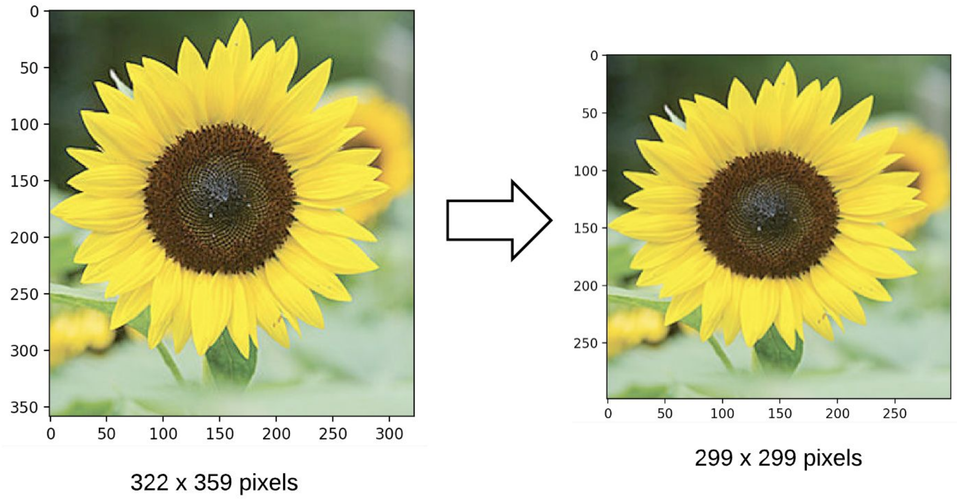

Figure 8 Image resizing using bilinear interpolation

3.2 Image Flip

In the image recognition problem, the image reversal should not affect the recognition results. So randomly flipped images are needed before training the CNN for image recognition. By preprocessing the image through this method, the trained model can better identify different angles of the object. No matter the flower faces to the left or the right, the classifier can accurately identify the same result. Figure 9 shows the images after using flip left right and flip up down.

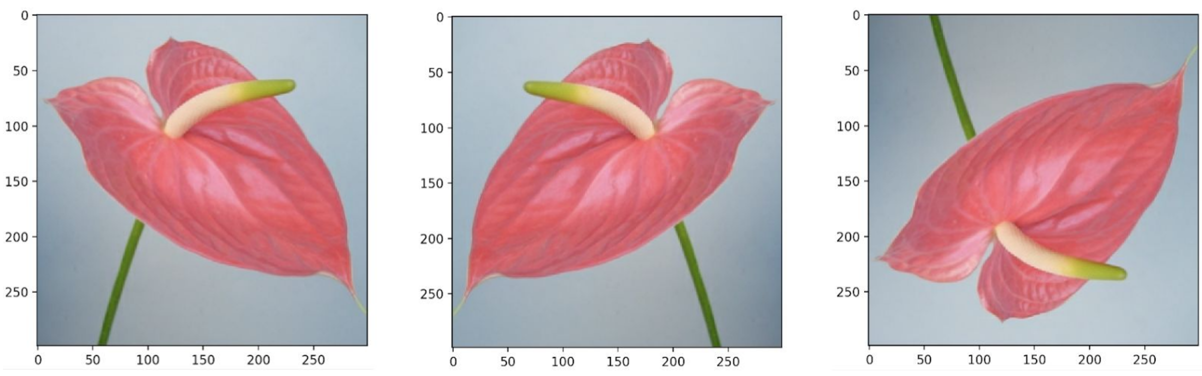

Figure 9 Image flip left right and image flip up down

3.3 Luminance & Contrast Ratio Adjustment

Similarly, in order to prevent the luminance and contrast ratio from affecting the final recognition results. Randomly adjusting the luminance and contrast ratio of the image in the dataset is needed. Figure 10 shows one example.

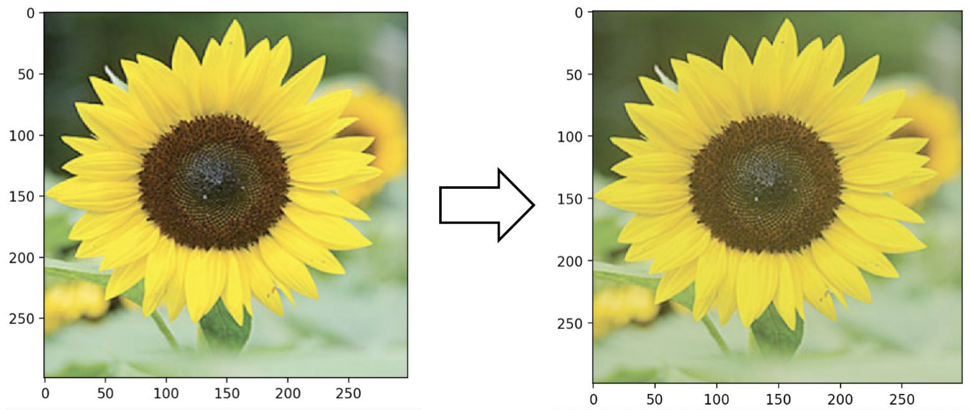

Figure 10 Image luminance & contrast ratio adjustment

3.4 Expanding Dataset by Image Preprocessing

The number of the image dataset crawled from the web is limited after purifying. However, a larger dataset helps improve the accuracy of recognition. By image preprocessing mentioned above, many training samples can be produced through one training image. Therefore, image preprocessing can be used to expand the dataset, and make sure the trained model can recognize the same flower of different size, orientation and luminance.

4 Applying Deep Learning

In early image recognition study, the biggest challenge is how to extract features, because image data unlike some other types of data that can be extracted by human understanding easily. Before the emergence of deep learning, SIFT, HoG and some other algorithm are usually used to extract good discriminative features and the method of traditional machine learning SVM algorithm are used to do the image recognition. However, the features extracted by these algorithms still have limitations, so the final classification results are not satisfactory. In addition, the accuracy of the image recognition through these methods is perennial difficult to break through. For this reason, the convolutional neural network of deep learning applied in this project will provide a new solution for image recognition problem.

Generally, through the deep learning method to solve the classification problem is divided into following steps:

1. The feature vectors of the selected entities are used as input to the neural network. For this project, the input is the original pixel of the preprocessing image.
2. Define the structure of the neural network. For this project, CNN was used as the architecture of the neural network.
3. Define the forward propagation algorithm.

4. Training the data to adjust the value of the parameters in the neural network. Also called training the model.
5. Using the trained neural network to predict unknown data.

4.1 Delinearization

The problem solved by the linear model is limited. Only through the linear change, even if the neural network structure is very deep, the expression ability of the neural network will not change. Therefore, deep learning needs to achieve delinearization to solve the linearly inseparable problem. And implementing a classifier of flowers is one of the linearly inseparable problems. Activation and biases are used to solve this problem. If the output of each node in the neural network passes through a non-linear function, the entire neural network model is no longer a linear model. This non-linear function is the activation. The main activation used in this project is ReLU function:

$$f(x) = \max(x, 0)$$

4.2 Loss Function

In the process of optimizing the parameters of the neural network, the loss function is used to measure the gap between the predicted value and the actual value. When the neural network is trained through the training set, the neural network can adjust the parameters through this gap value so that the output vector of the training set will closer to the expected vector. The main loss function used in the project is the cross-entropy. This loss function can be used to describe the classification precision of the model to the problem. The smaller the loss, the smaller the deviation between the classification result and the true result of the model, that is, the more accurate the model. By continuously reducing this loss, the parameters in the model can reach a global optimum solution or local optimal solution.

4.3 Applying Convolutional Neural Network

Deep learning does not need to separate the feature extraction process and classification training process, it automatically extracts the most effective features when training the model. CNN has weight sharing structure, it significantly reduces the amount of parameters in the neural network. Therefore, CNN reduces the complexity of the neural network model and alleviates overfitting problems to some extent, also improves the training efficiency. The concept of convolutional neural network originated from the receptive field[7] at first. At that time the study of cat visual cortical cells found that each visual neuron would only deal with a region of the visual image, that is, the receptive field. This kind of neural network structure is very suitable for training image data. In the convolutional neural network, the first convolutional layer can directly accept the input of the image pixel level (e.g. a JPEG image in RBG colour of 299*299 is 299*299*3). Each convolutional operation handles only a small piece of the image. Each convolutional layer extracts the most efficient features of the data. This method can extract the most basic features of images and then combine and abstract them into more complex features.

4.3.1 Structure of CNN

The CNN in this project has the following 5 structures:

1. Input layer

The input layer is the input of the entire neural network. It represents the pixel matrix of the image. In this project, all the flower images are converted into a three-dimensional matrix. The length and width of the matrix represent the size of the image (299*299). And the depth of the matrix represents the color channel of the image. In RGB color mode, the depth of the image is 3. Therefore, the size of the input matrix is 299*299*3 in this neural network.

2. Convolutional layer

The convolutional layer is the most important layer in CNN. The input of each node in the convolutional layer is only a small piece of the last layer of the neural network. This kind of convolutional operation is called the filter or the kernel. The convolutional layer attempts to deeply analyze each small piece of the neural network to obtain more complex features. The size of the filter, padding mode and the stride value used in this project have many types.

3. Pooling layer

The pooling layer is used to reduce the size of the matrix. The pooling layer can be thought of converting a higher resolution image into a lower resolution image. Therefore, the parameters in the neural network can be further reduced through the pooling layer. Max pooling and average pooling were used in this project.

4. Fully connected layer

After through many convolutional layers and pooling layers, the final classification results are given through the fully connected layer. For the 50-classification problem, there are 50 outputs. For the 125-classification problem, there are 125 outputs.

5. Softmax layer

The softmax layer can turn the final results into a probability distribution results. Following formulas shows how it works:

$$soft\max(x) = normalize\left(\exp(x)\right)$$

$$soft\max(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

The CNN architecture used for this project refers to four classic neural networks, they are AlexNet[9], VGGNet[10], Google Inception Net[8] and ResNet[11]. Mainly reference to the structure of Google Inception. Although Inception structure is deeper than AlexNet, the parameter number of Inception structure is much smaller than AlexNet. So training the model with inception structure is more efficient and reliable. Figure 11 shows the structure of the whole convolutional neural network in this project. 3x3 small size kernel are used in the first three convolutional layers of this structure. The essence of the structure of VGGNet is like this. After that, 1x1 size kernel is used to combine the features cross the channel in low cost. Two max pooling layer using 3x3 size kernel to further compress the size of the input image data. Therefore, the first seven layers of this neural network can compress the image size and abstract the image features. After that, 11 modules consisting of several convolutional layers and pooling layers are used to further compress the image size and combine more complex features. The module in the CNN structure in this project have five main structures (see Figure 12). All of the modules have the 1x1 size kernel because the 1x1 size kernel is very cost-effective, it can add a layer of feature transformation and achieve nonlinearity with very small calculation. Moreover, the design of these modules in the neural network also based on the Hebbian principle: "Cells that fire together, wire together". For image data, the data for the adjacent area is more relevant so that the adjacent pixels should be connected together by the convolutional operations. The kernel of 1x1 size can correlate these correlations in the same spatial location but with different channel features, so this is also why the 1x1 size kernel is frequently used in the module. The correlation of the nodes connected by 1x1 size kernel is the highest, but some other sizes of kernel like 3x3, 5x5 or 7x7 are also high. By adding these bigger convolutional kernel in the module, the diversity of the modules can be increased. At the same time, through the combination of different sizes of kernels of convolutional layers and pooling layers, a high efficiency to meet the sparse structure of Hebbian principle can be built. Then, the performance of the whole neural network to process the image data can be greatly improved. In this structure, replacing the 5x5 size kernel with two 3x3 size kernel was used to reduce the number of parameters and overfitting problems. Moreover, splitting a larger two-dimensional kernel into two smaller one-dimensional kernels is also useful. For example, NxN size kernel can also change to 1xN size kernel plus Nx1 size kernel. It can achieve the same effect while speeding up the calculation, and add a nonlinear feature transformation ability to the module.

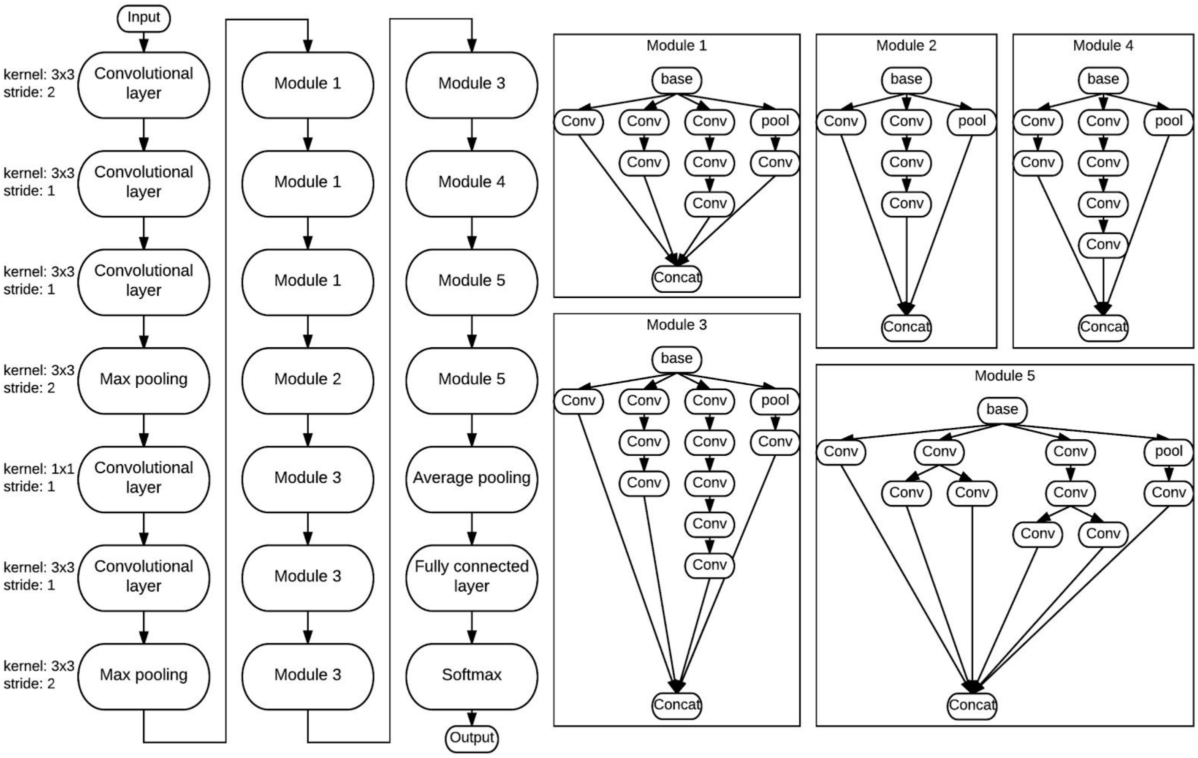

Figure 11 CNN structure

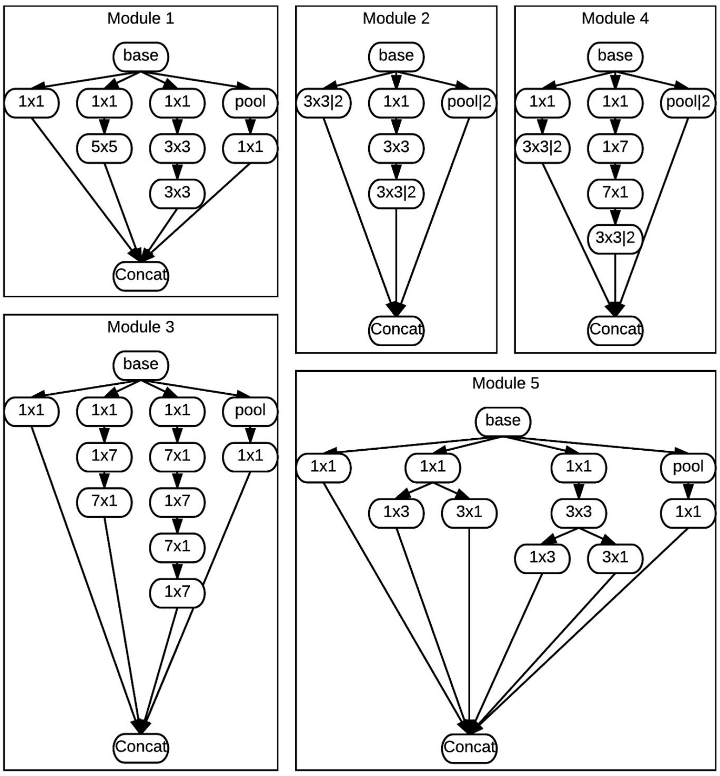

Figure 12 The kernel size in the module (|2 means stride is 2, default is 1)

4.3.2 Training Process

Only the trained neural network can really solve the classification problem (flower recognition problem). In this project, 80 percent of the dataset was used as the training set, 10 percent of the dataset was used as the validation set and 10 percent of the dataset was used as the test set. Figure 13 shows the Training process (back propagation). As can be seen from Figure 13, The training process is an iterative process. Before each iteration begins, select a part of the training data as a batch. Then, using forward propagation algorithm to get the prediction results of the neural network model. Since the training data already has been correctly classified, the gap between the predicted classification and the actual correct classification can be calculated. Finally, through this gap, using back propagation algorithm updates the value of the neural network parameters, so that the prediction results of the neural network model on this batch can closer to the actual answer. In this way, the neural network will be able to classify flowers well after a massive time of training. In this project, the model was trained about 20 thousand times at first. After analysis, overfitting problem was found after 10 thousand times training. Therefore, training step was set to 12 thousand at last.

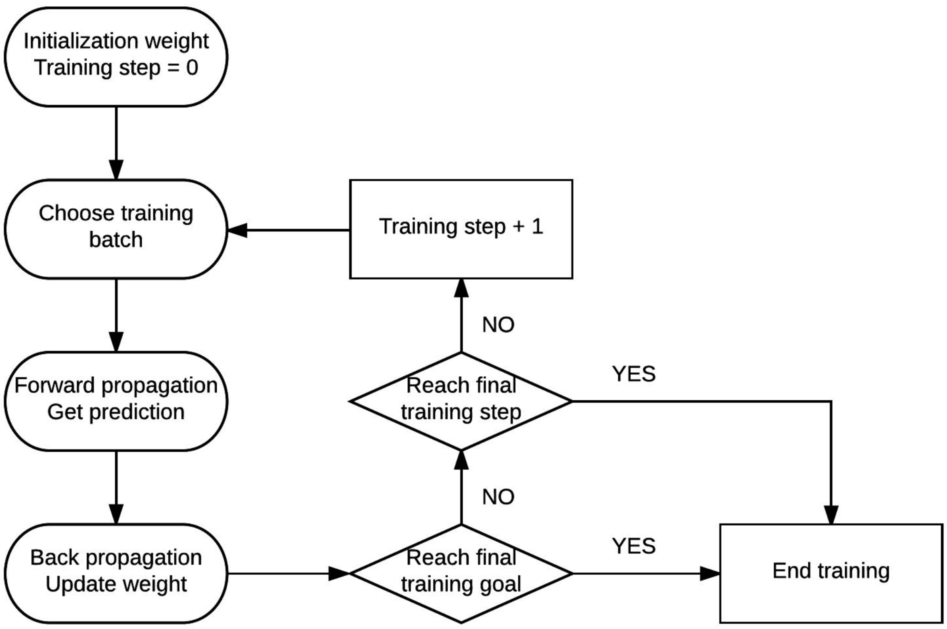

Figure 13 Training process

4.3.3 Optimization Algorithm

In deep learning, the algorithm of optimizing parameters usually use gradient descent algorithm and stochastic gradient descent algorithm. But the gradient descent algorithm will take too much time for the calculation and can not reach the global optimum solution. Because it minimizes the loss of all the training set, the loss function on all training set is calculated in each iteration. In the massive data, it is very time-consuming to calculate the loss function of all training set. However, using stochastic gradient descent algorithm can speed up the training process. This algorithm does not optimize the loss function on all training data, but in each iteration, it randomly optimizes the loss function on a training data.

In this way, the speed of each round of parameter updates is accelerated. But the loss function on a certain data become smaller does not mean that the loss function on all data become smaller, so the neural network trained by stochastic gradient descent algorithm may not even reach the local optimal solution. In order to combine the advantages and disadvantages of these two algorithms, in this project, a small number of the loss functions of the training data are calculated in each iteration. This small part of the data is called a batch. By this method, each time optimize the parameters of the neural network in a batch does not much slower than optimizing the parameters for a certain data. On the other hand, when batch is used, the number of iterations required for convergence can be greatly reduced, and the results are closer to the effect of gradient descent.

4.6 overfitting problem

The model trained by the CNN is designed to recognize unknown flowers in real life. Therefore, the training model can not be too simple or too fitting. If the model is too simple, the prediction accuracy of the model will not be good. If the model is too fitting, the prediction results will have a good effect on the training data set but perhaps a bad effect on the unknown data due to the existence of the noise. So a model that will not pay too much attention to the noise and can commendably depict the trend of the problem is needed for the project. To avoid overfitting problems, in addition to setting a reasonable step of training, the regularization method can be used. Regularization method is to add indicators which describe the complexity of the model to the loss function. The L2 regularization was used in this project:

$$R(w) = \Sigma_i \left|w_i^2\right|$$

4.7 Migration Learning

Due to the limited time of the project, it is impractical to constantly adjust the training dataset and constantly completely retrain the neural network. Therefore, migration learning method was used to quickly adjust parameters through a new training set based on a neural network model trained on the previous training set. Through the migration Learning, the parameters of all convolutional layers in the previously trained model can be retained. In general, in the case of the training data is big enough, the effect of the migration learning is not as good as the completely retraining, but the training time and the number of the training samples required for the migration learning are much less than retraining a new model. In the later stage of the project, this method can be used to facilitate multiple training dataset tests.

5 Results and Analyzation·

5.1 Training & Validation Accuracy

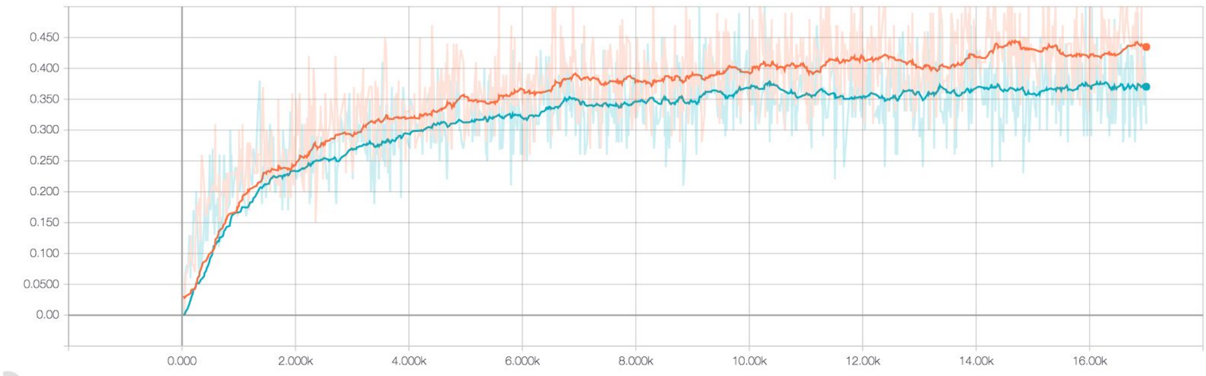

Figure 14 125 flower species classification accuracy

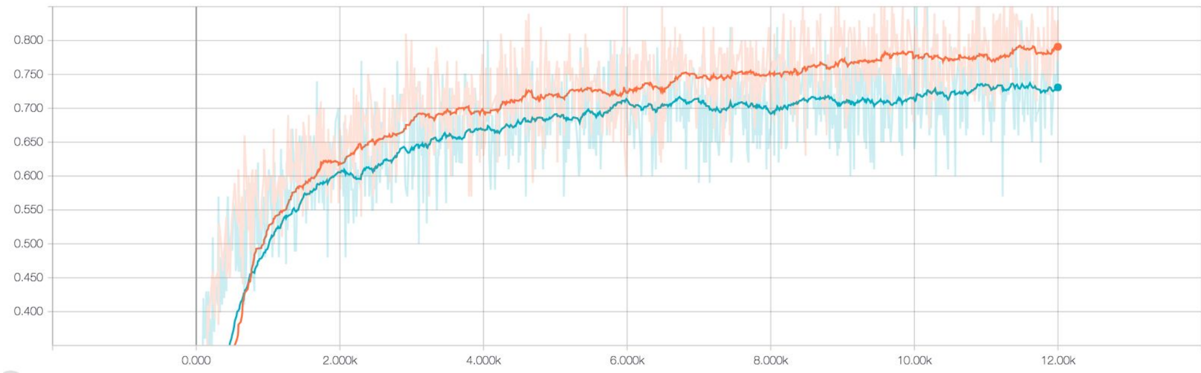

Figure 15 50 flower genera classification accuracy

For two case studies: 50 flower genera classification and 125 flower species classification, Figure 14 and Figure 15 shows the rising trend of training and validation accuracy where orange line presents training accuracy and blue line presents validation accuracy. As both graphs showed, before 12000 times of iteration, train/validation the accuracy keeps increasing. After 12000 times of iteration, the rising trend of accuracy goes smoothly, so for the purpose of preventing overfit problem both case studies use 12000 times iteration to train the model. As the graphs show 50 flower genera classification can reach around 70% accuracy and 125 flower species classification only reach around 40% accuracy. This is the main reason why there are two case studies in this project: different flower species belong to same genus may have very similar appearance to learn features and make it extraordinary hard to differentiate each other correctly.

5.2 Predict Precision

After finished purification procedures in the dataset, around one-fifth data are selected and moved into another directory and only four-fifths data left in the original set is used to training a model. This one-fifth data was used to test the predict precision for the model. Figure 16 shows the success percentage of prediction result of 50 flower genera classification case study for each genus:

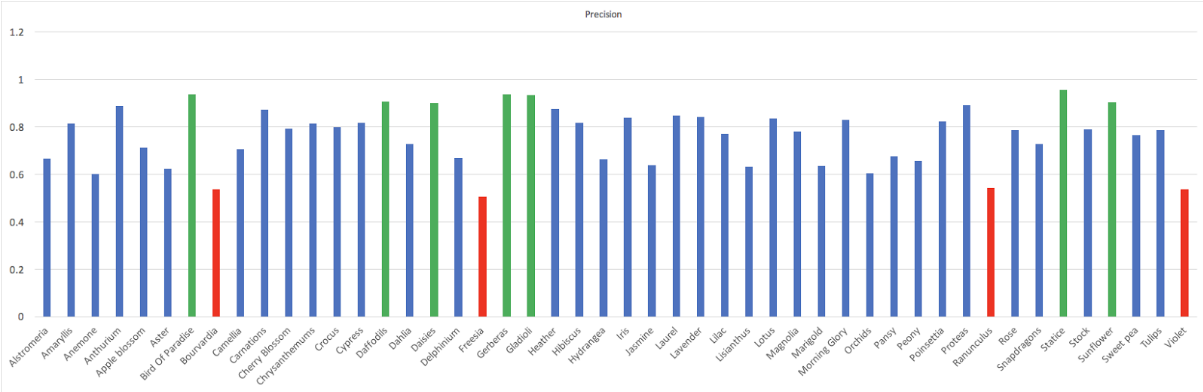

Figure 16 50 flower genus classification predict presicion

The average number for predict success percentage is 76.2%, any green column means this flower genus set has more than 90% images are predicted correctly in test dataset (one-fifth total purified dataset), any red column means this flower genus set has less than 60% images are predicted correctly in test dataset (one-fifth total purified dataset) and blue columns present for other sets.

5.3 Analyzation

According to Section 5.2, different flower genus/species has different prediction success percentage, some have higher percentage even more than 90% and some have lower percentage only 50%. All these sets have been purified by purification procedure introduced in Section 2, the reason why different sets' prediction success percentages have such a difference has been listed below:

high prediction success percentage:

1. Unique Appearance: Shape and color are most important features to recognize images, a flower with a unique appearance can easily being identified because its shape and color contain great weight during classification period while model is running.
2. Pure Dataset: After purification, usually there will be some wrong images left in the dataset, these images may not affect too much to the prediction success percentage but if a set has no wrong image in it then it can make the classifier make a better decision in this set.
3. Correct Photograph Angle: An object can be taken the photo in several angles and for most objects photographs taken by different angle can be very different with each other like a flower photograph only contain petal or a flower photograph contains several same kinds of flowers. If the image needs to be classified is being taken by the correct angle like showing the whole flower including petal, rachis, bud and some on then it will be much easier to recognize its flower species.

low prediction success percentage:

1. Similar Appearance: Like introduced in Section 2.1, once the training dataset is not big/strong enough, it will be hard to differentiate flowers in the same genus because their appearance looks like each other.

2. Different Weight: Some flower images can "survive" purification but not helpful for final classification that is because during the purification process there is some features have great weight and these weight can help itself get a pure dataset after purification but after that during building classification model process these features' weight become small and helpless and cannot differentiate well with other flowers.
3. No Similar Photo In Dataset: If there are no previous images being trained in the dataset like there are only mature flower images in the database and once it needs to classify an immature flower image then the model will make the wrong prediction.

6 IOS Application Implementation

This section introduce the IOS Application implemented in this project using Tensorflow and classifier model to recognize flowers in real-time.

6.1 Interface Design

For the implementation of the mobile application, the interface design is very important. Because of the limited screen size of the mobile phone, an excellent visual interface and a good operating experience for users are needed. In this project, minimalist approach and easy use approach were used as the core of the interface design. There are two main views in this application, prediction view and category view. Users can recognize the unknown flowers in the prediction view and use the category view to check the flower list the application support to recognize. The prototype design of the interface is shown in figure 17.

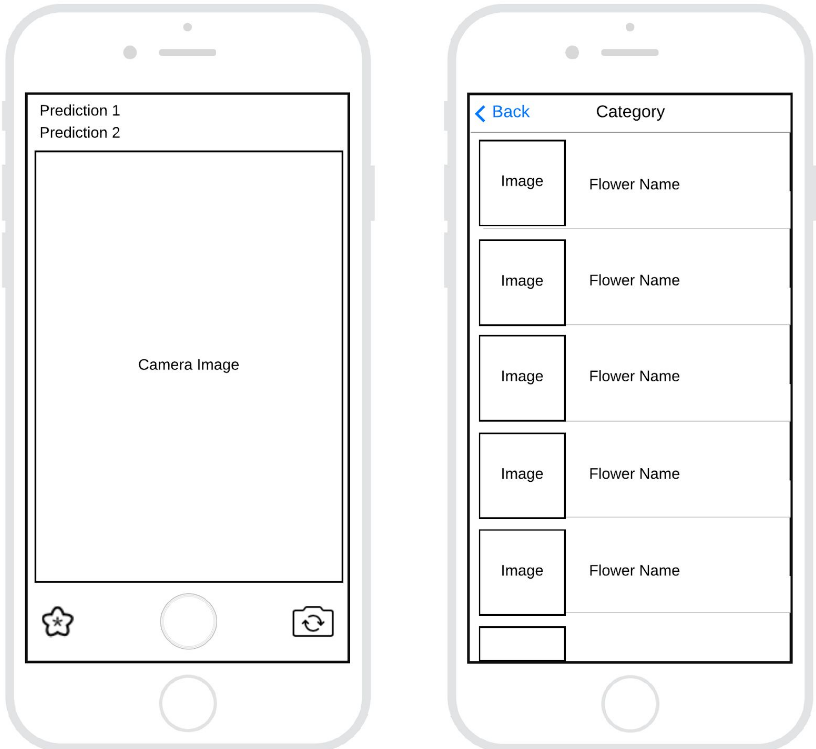

Figure 17 knowFlower interface design

The actual interface is shown in figure 18.

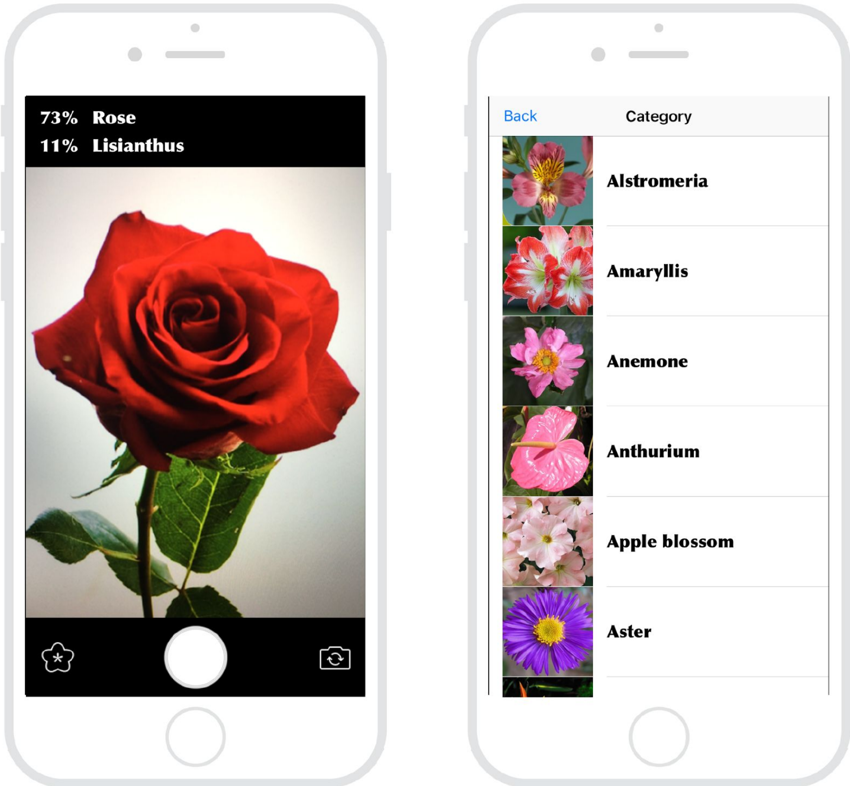

Figure 18 knowFlower actual interface

6.2 Model Optimization

IOS devices have limited amounts of memory and limited calculation ability. The IOS version of the Tensorflow library only supports for operations that are common in inference and do not have large external dependencies. Therefore, the classifier model trained previously needed some optimization to be used in the IOS devices. A Google's build tool Bazel was used to optimize the classifier model. The model optimization has the following steps:

1. Remove operators that are not supported by IOS
2. Round the weights of the model within a constant to 256 levels
3. Do memory mapping

After these steps, an optimized model that can run on the IOS device was built. The model will not occupy too much storage space in IOS device because it can be compressed much smaller than before. The RAM in IOS will not have a lot of pressure because of the memory mapping. Most importantly, the precision of the classifier model only reduces by a small amount (less than 0.5% drop).

6.3 Camera Interface

The Camera Interface is also the homepage in this application. It contains five main parts:

1. At the top of the screen, it shows at most two flowers' name with the highest predict percentage which at least needs to be five percent.

2. At the middle of the screen, it shows the image captured by the camera, because the algorithm allows the application to classify flower in real-time so to use this application just need to aim the camera at the target object and if user do not like the angle of the camera they can change it at any time.
3. At the left of the bottom is a button looks like a cute flower, this is the jump button used to jump to the category interface to see all the predict available in this application.
4. At the middle of the bottom, a big white square button, once it was touch the Camera Interface will be frozen until jump to Category Interface, retouch this button or use the turnaround button.
5. At the right of the bottom is a turnaround button, it can switch the camera between the front-facing camera and rear camera.

6.4 Category Interface

Category Interface represents all recognizable flowers by this application, it shows as a traditional drop-down table and each row of the table shows an image of a flower at left and the name of the flower at the right-hand side of the image.

7 Future Work

The next aim of this IOS application is to recognize more flowers with a better prediction. So the future work can be divided into four parts:

1. Build purer and larger dataset, help training model with better predict precision.
2. Improve the structure of the convolutional neural network, try some better structures to make the training process more efficient and more reliable.
3. Use meta information to classify flowers, using geometry, height, width and some other information to do a pre-classify process to delete some impossible result.
4. Adding more recognizable flowers into this application.

8 Conclusion

In this project, an IOS application was implemented to help people recognize flowers in the real life. This report discusses the relevant techniques and methods applied to implement this application. It includes data mining, dataset processing, image processing, deep learning method, IOS application development and result analysis. By these methods, the identification ability of software can finally close to the level of manual annotation.

References

1 All products require an annual contract. Prices do not include sales tax (New York residents only). (n.d.). Number of smartphone users worldwide 2014-2020. [online] Statista. Available at: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/
2 Tutorials | TensorFlow. (n.d.). [online] TensorFlow. Available at: http://www.tensorflow.org/tutorials
3 Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M. and Ghemawat, S., 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.
4 McCulloch, W.S. and Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5(4), pp.115-133.
5 Burnley Plant Guide Online. (n.d.). [online] BPG Online - Access denied. Available at: https://bpg.unimelb.edu.au/qs-results.jsp?id=4&advanced=true&bpg-search=&bpg-botanical-name=&bpg-family=&bpg-common-name=&bpg-description=&group=wild&type=bulb,+corm+or+rhizome&submit=Find+Plants
6 Ahmed, A. (n.d.). 50 Most Beautiful Flowers In The World. [online] STYLECRAZE. Available at: http://www.stylecraze.com/articles/most-beautiful-flowers-in-the-world/#gref
7 Sherrington, C.S., 1906. Observations on the scratch-reflex in the spinal dog. The Journal of physiology, 34(1-2), p.1.
8 Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).
9 Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
10 Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
11 He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).
12 Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A., 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261.
13 Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2818-2826).

14 Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T., 2014, June. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In Icml (Vol. 32, pp. 647-655).

Appendix

Source code

https://github.com/HeavenMin/PlantImageRecognition

Video Demo

https://youtu.be/zXOIiAQ-4GI