# Flowers Identification with CNN

## Executive Summary

The task aims at training a deep neural network to classify common flowers in HK. The source photos come from google images. Resizing and augmentation are performed to standardize and increase input samples. A convolutional neural network (“CNN”) model is trained initially, followed by varying parameters in order to reduce overfitting and increase accuracy. In the end, a 3-convoution layer CNN works well for this task, with accuracy on test samples at 91%, improved from 63% in initial model.

## Description of Task

We often see flowers but do not know its name. It is useful to develop a simple neural network to classify common flowers. It may be useful to classify common flowers during hiking. The main procedures include data preparation and model training.

## Data Preparation

### Data collection

I searched online and found many common flower types in HK. For simplicity, I only chose 3 classes for this exercise. I searched on Google image and randomly downloaded 50 images for each class.

1. 醉蝶花 Cleome spinosa Jacquin
2. 紫玉蘭 Magnolia lilifora
3. 蔓性野牡丹 Dissotis rotundifolia（Smith）Triana

### Resizing and data augmentation

Given that all the images have different dimension, I resized them into a fixed dimension (50x50 pixels). Additionally, I performed a data augmentation process to double the number of images per class from 50 to 100. Below is an example:

It is observed that the resized image has a little distortion because the original image is not a square in size. The cropped image has proper shape but looks like an enlarged image. We hope the CNN can handle both instances.

Sample images:

|  |  |  |
| --- | --- | --- |
| Original (275x183) | Resized (50x50) | Cropped image (50x50, cropped 100x100 pixels from center, then resized) |

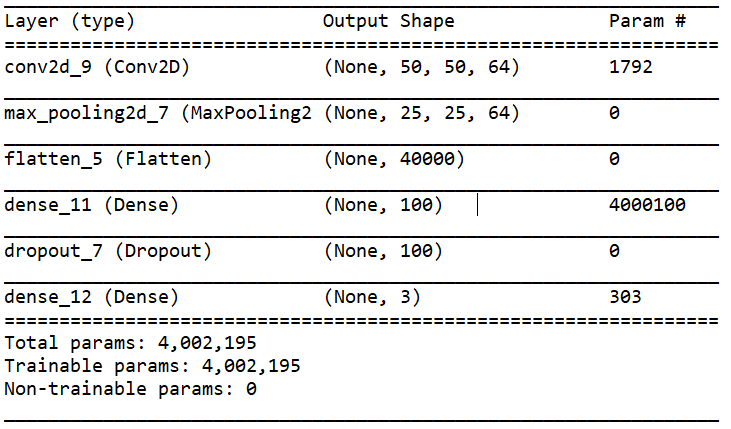
### Database preparation

The images are converted into numpy arrays and stored into an .npz file, with 80% of samples for training and 20% for testing.

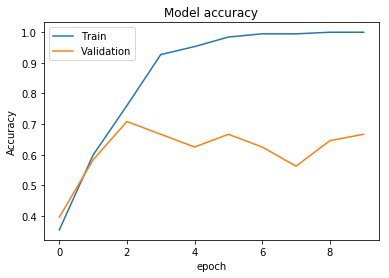
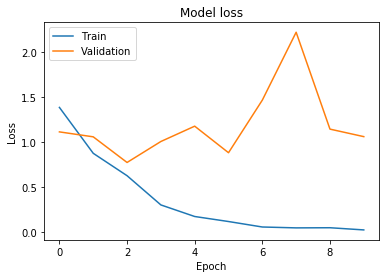
## Structure of CNN

I started with a simple CNN structure with details below. The problem with this initial model is the apparent overfitting after epoch 2, as training loss continued to decline while validation loss started to increase. The accuracy for test set is **63%**. I treated this as a benchmark for future comparison.

Model structure summary:



Training result:



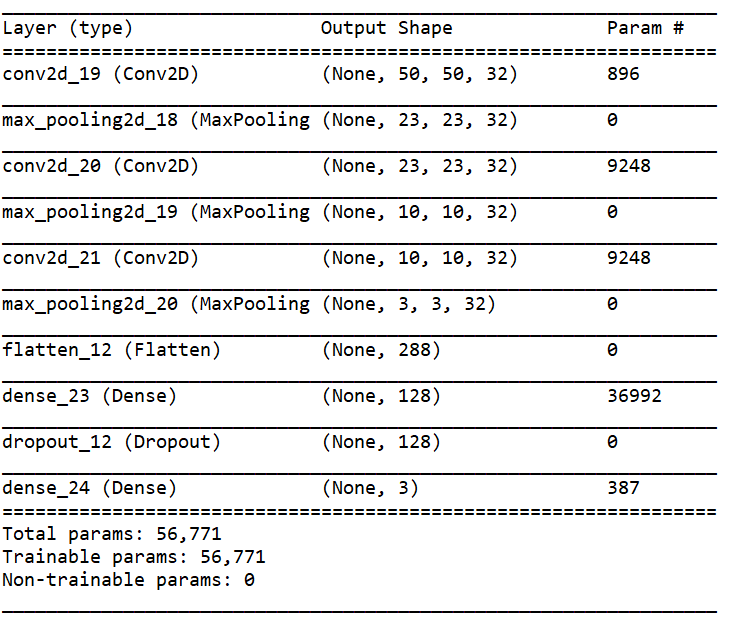
Test set accuracy: 0.633

## Parameters for Variation

Efforts were made to tackle overfitting and low accuracy issues in the initial model. Below are the major changes and rationale:

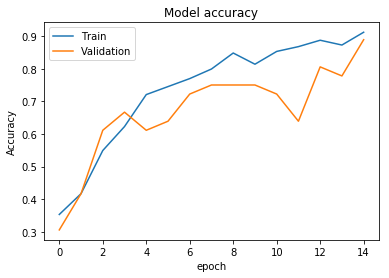
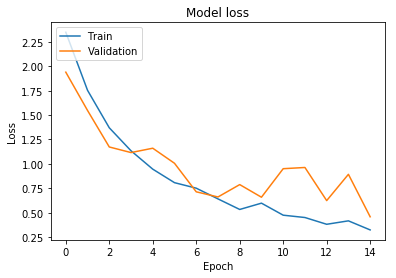
|  |  |  |
| --- | --- | --- |
| **Area** | **Parameters change** | **Rationale** |
| # of convolution layers | 1 convolution layer -> 3 convolution layers (with max pooling) | There may be higher level of features that can be detected after max pooling, so additional convolution layers should be helpful. |
| Max pooling | Kernel size 2->5 | Detect larger features |
| Learning rate | 0.01 -> 0.001 | The original decline of loss appears too fast at the beginning, a smaller learning rate is appropriate. |
| Regularization | Added a l2 regulariser at the dense layer (before fully connected layer) | Reduce overfitting |
| Validation split | 0.2-> 0.15 | Increase number of training samples |
| Epoch | 10->15 | Increase training iterations |

Below is the modified model structure:



## Results discussion

The final training / validation / testing accuracy are 0.9118 / 0.8889 / 0.9166 respectively. The testing accuracy sharply improved **from 63% to 92%** in the modified model. The training and validation loss decline is smoother and overfitting problem is not apparent.



## References

Green Touch 香港常見樹木園藝生活

<https://www.treehk.com/>

Tutorial: Overfitting and Underfitting

<https://keras.rstudio.com/articles/tutorial_overfit_underfit.html>

## Table of Summary

|  |  |
| --- | --- |
| Classification task | Identification of common flowers in HK using CNN |
| Objective of investigation | To increase model accuracy |
| Image size, colour or grayscale | 50x50 pixels, color |
| Reference/source of the images | Google images |
| Number of classes | 3 |
| Available Images per class | 100 |
| Number of training images | 240 |
| Number of testing images | 60 |
| Structure of CNN | 3 layers of convolutional plus max pooling, followed by a dense layer, then fully connected layer |
| Parameters of variation | # of convolution layer, max pooling kernel size, learning rate, regularization, validation split, epoch number |
| Training time | Around 2 minutes |
| Testing accuracy | 92% |
| Software platform | Keras |
| Hardware platform | Lenovo X1 Carbon Laptop |

Sample images:

|  |  |  |
| --- | --- | --- |
| Class 0 | Class 1 | Class 2 |