Classify Flower Species Using CNN - Hong Xu, Lufei Wang, Zihao Li, Tianjing Cai

The goal of our project is to apply a set of Convolutional Neural Network (CNN) models, including VGG16, ResNet50, Inception v3, ResNet152, and VGG19 to classify 102 flower categories

Introduction & Problem Statement

Our team want to compare the performance of each CNN model and evaluate the effect of image augment to the results. Top 1, Top 3, and Top 5 accuracy represent the rate of each CNN model that contains top1, top 3, and top 5 highest possibilities for predicting the correct flower species.

Related Work

In a blog (Github: Gogul09), a similar study showed that the Inception v3 scored the highest Top 1 and Top 5 accuracy on a dataset that contains 17 flower species. Our team will run CNN models on a larger dataset and consider the effect of image augment to the accuracy.

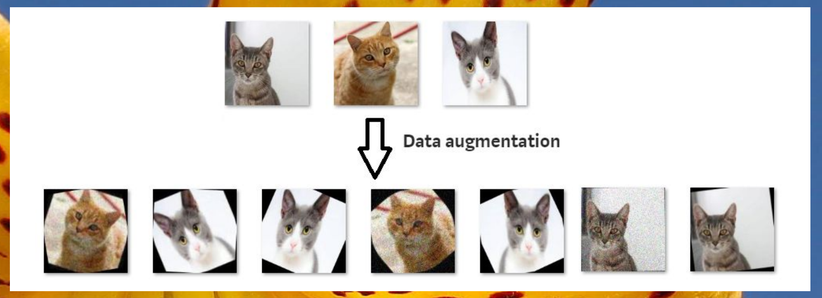
Dataset

The Department of Engineering Science at the University of Oxford have created a flower dataset, consisting of 102 flower categories. Each class consists of between 40 and 258 images. Some flower species may have large variations within the same category.

For this project, our team used 6149 images as training set, 1020 images as validation set, and 1020 images as test set.

Image Augmentation

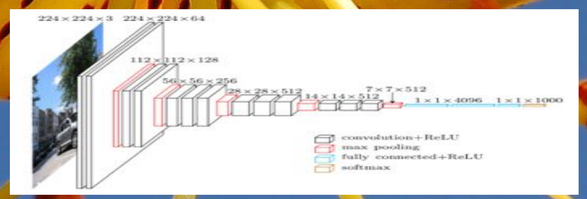
Randomly applying various transformations to your inputs while training models to enlarge the training data and prevent overfitting.



Models

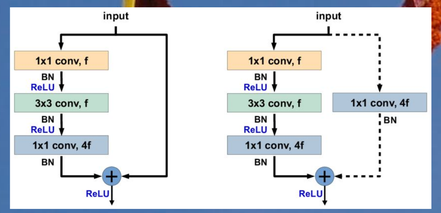
VGG:

VGG use only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Two fully-connected layers, each with 4,096 nodes are then followed by a softmax classifier (above). The 16 and 19 stand for the number of weight layers.



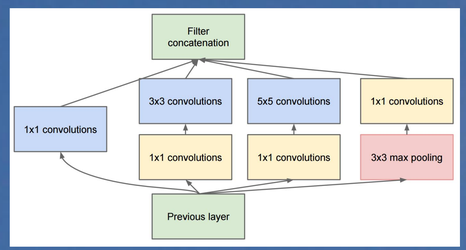
ResNet:

ResNet model relies on micro-architecture modules , which is a collection of micro-architecture building blocks (along with your standard CONV, POOL, etc. layers) leads to the macro-architecture.



Inception:

The Inception module aims to act as a “multi-level feature extractor” by computing 1×1, 3×3, and 5×5 convolutions within the *same* module of the network — the output of these filters is then stacked along the channel dimension and before being fed into the next layer in the network.



In keras, these pretrained models and the weights are available. Our team resized the images to the required size of different models and used our data to update the weights. Early stopping based on the validation set is used to prevent overfitting.

Results

	Top1 Accuracy	Top3 Accuracy	Top5 Accuracy
ResNet50	0.5951	0.7902	0.9324
ResNet50 (NO Image Augmentation)	0.5656	0.7617	0.9019
VGG16	0.6549	0.8088	0.9176
Inception_v3	0.9078	0.9608	0.9922
ResNet152	0.6326	0.7924	0.9421
VGG19	0.7268	0.8552	0.9125

Based on the result, the image augmentation could improve the prediction accuracy by 3%. With the image augmentation, all CNN models score at least 60% prediction accuracy. Among all the CNN models that tested with the test set, CNN with Inception v3 model scored the highest accuracy where the Top 5 accuracy is around 99.22%.