MsBD 6000b assignment 2

Introduction

In this project, I am going to use a deep network model to classify flower images amount of 5 classes. I will try to compare different type of model with different type of image preprocessing method.

Model

There are 3 models I am going to try out, they are 1) standard convolutional network, 2) multi-towers convolutional network with different kernel size and 3) pre-trained inceptionV3 network with additional Dense output layer.

Data Pre-processing

There are many preprocessing method for image classification, I will try to have 3 different method to compare their difference.

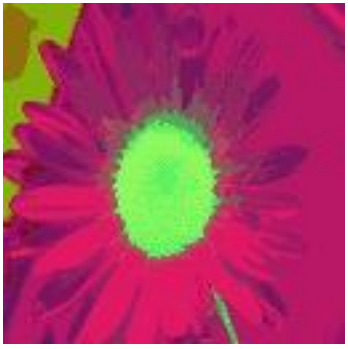
1) Baseline: Original image and standardize the dimension.
2) HSV color filtering and standardize the dimension.
3) Image with Edge detection filtering and standardize the dimension.

Finally, I will apply ImageDataGenerator for keras in order to enhance our dataset.

Test result on different pre-processing method

I will compare the performance between 3 types of preprocessing method. The model I use to test is traditional convolutional network with 3 epochs.

The first method is HSV color filtering. The reason I choose HSV color filtering is it possibly enrich the color of flower. Because most of the flower has high saturation. So I can imagine that the image convert to HSV color space can enhance the characteristic of flower.



The second method is edge detection filtering. Simply using edge detection is because I would like the model to learn the outline of the flower. No matter what the color is, the model can extract the feature of flower shape. Here is the example of the image.

Finally I will only resize the original image as baseline.

The result is very interesting, original image has the best performance. I can imagine that the size of the flower in many image are too small such the it is hard the learn the shape of flower by only the edge. And also the HSV color space makes all flower classes having similar color tone. It could be the reason why it perform bad.

Model
The setting I applied to all 3 models are the same:
Input size: 128*128*3
Epoch: 50
Learning rate: initial 0.01 (with Reduce learning rate when a metric has stopped improving.)
Filter size of kernel: 5,5
Batch size: 128
Optimization: Adam

Result

	Validation Accuracy
Standard Convolution NN	~71%
Multi-tower CNN	~68%
InceptionV3 pre-trained	~81%

Conclusion
The deepness and the size of the CNN is the key of good performance. The reason of picking pre-train model with additional layer is it reduce a lot of training time. I did build up VGG16 model for training. However, the training time is huge and it is impossible to be completed in few days.
Secondly, the input size of the image is also very important to achieve good result. By double up the dimension with the same kernel size, there was more than 10% accuracy gain but also consume a lot of training time.

Finally, I found that ImageDataGenerator is a good tool to avoid over fitting. Since it can generate a lot of image augmentation from the same set of data. And sharply increase the validation accuracy along with the training accuracy. However processing power is one of the problem of the project. Without GPU, laptop computer is too slow for deep network training.