Project 2 Report

Wang Chengming 20450392

November 21, 2017

1 Data Preprocessing

In this project, a huge dataset of flowers is given for us to do classification using deep models. The training and validation dataset contains 3119 flower samples. Each of them belongs to one of five species. The first step is to load these pictures and transform them into numpy arrays for further processing. We can use *imread*, *imshow* function from scipy package to transform and show the pictures. Below are the results.

```
In [9]: plt.subplot(1, 3, 1)
        _ = img_norm(img_li[0], img_label[0], False)
        plt.subplot(1, 3, 2)
        _ = img_norm(img_li[1], img_label[1], False)
        plt.subplot(1, 3, 3)
        _ = img_norm(img_li[2], img_label[2], False)
```

The image size is $(333, 500, 3)$. Three dimensions are height, width and depth. Depth is just the RGB. It would be computational expensive if we keep the original size of images and feed them into the model. So we need to resize them to save computation time and memory.

2 Model Building

My first thought is to build a convolutional neural network model by myself, following the structure of VGG16. I built a simlified version of it, with fewer neurons on each CONV and fully connected layer. But the training speed and accuracy are not satisfying. So I turned to some pre-trained models. The advantage of pre-trained models is that there's no need to backprop through so many layers on my laptop. The weights have already been trained and stored properly. Also, these models have shown their power on Imagenet dataset. They are very likely to do well on this dataset.

The model I tried to use is ResNet50. Since ResNet swept all classification and detection competitions in 2015, I believe it will achieve great performace on this dataset. From keras.applications I downloaded the weights of ResNet50 and fine tuned it by adding another dropout layer with 60% neurons dropped in order to prevent overfitting. The output layer contains five neurons, denoting five species of flowers, and softmax activation. The model was trained for 50 epochs with batch size 64. The validation accuracy is around 89%.

```
In [ ]: from keras.applications import ResNet50

        base_model = ResNet50(include_top=False, weights='imagenet', input_tensor=Input((224, 22

        for layer in base_model.layers:
            layer.trainable = False

        x = base_model.output
        x = Flatten(name='flatten')(x)
        x = Dropout(0.6)(x)
        # x = Dense(64, activation='relu', name='fc1')(x)
        x = Dense(5, activation='softmax', name='predictions')(x)
```

After the model was trained, I tested it on the validation set. It reached accuracy of 90%. The final step is then to test on the test set and output the results.