Chapter 1

Introduction

1.1 Definition of the problem

1.2 Why problem is worth solving

1.3 Summary of solution

Chapter 2

Literature Review

CNNs, where they came from, who invented them, imagenet challenge

Chapter 3

Classification

3.1 Overview of classification pipeline

Photo of a flower is taken by the user. CNN converts the photo into a characteristic vector. SVM classifiers used to determine the species of flower.

3.2 Convolutional neural network

Used to generate characteristic vector from an image. Treated as a black box in this project.

3.3 Support Vector Machine

3.3.1 How SVMs work

3.3.2 Training and testing SVMs

3.3.3 SVM accuracy

3.3.4 Improving SVM accuracy

Mirroring

Sampling (Jittering?)

3.3.5 How SVMs are used in the classification pipeline

3.4 MatLab classification script

3.4.1 Explication of classification script

Script uses the CNN to generate a characteristic vector from an inputted photo. We test that vector against the 102 weight vectors found during training to produce a classification prediction.

Chapter 4

Client architecture

4.1 Overview of client architecture

Android application consists of three activities, or screens with which the user interacts. The Main Activity allows the user to take a photo or choose a photo from gallery to upload. The photo is uploaded to the server, which classifies the flower, and returns the result to the Results Activity. The results activity displays the eight top classification results, and allows the user to click on each, and find out more information in the Detail Activity.

4.2 Main Activity

Main Activity consists of two buttons, which launch the camera and gallery image picker respectively.

4.2.1 Camera intent

4.2.2 Gallery image picker intent

4.3 Results Activity

Description of code used in the Results Activity

4.3.1 Uploading the photo

AsyncTask (keeping slow processes such as web connections off the UI thread)

Connecting to the server

4.3.2 Loading classification results

Parsing JSON Array

ListView

Picasso image downloader library

4.4 Detail Activity

Description of code used in the Detail Activity

4.4.1 TODO after Detail Activity implemented

Chapter 5

Server architecture

5.1 Overview of server architecture

5.2 Flask server

5.2.1 How server accepts the photo

5.2.2 Error catching

Safe filenames

Allowed filetypes

5.3 Connection between servers

5.4 Backend server

5.4.1 Description of how server works

5.4.2 mlwrap

Chapter 6

Conclusions and extensions

6.1 Conclusions

6.2 Porting algorithm to Android