Portable Flower Recognition Device

Group 14 -- Qichao Gao, Zekun Wei, Tianying Zhou
TA–Nicholas Ratajczyk

Sept. 20th. 2018

Table of Contents

1 Introduction

1.1 Objective

There are many different types of flowers, and they are all so beautiful but at same time they all look the same for many people who are not experts. The difficulty of recognizing flowers is depending on the geometric shape, texture, color and various parts from the flower. Recognizing different flowers correctly is a difficult task for people to learn and master, and people don't have to learn it since they don't need it for their daily life. But it would be really cool and sometimes useful to have this skill, for example, you can identify which of the flowers are edible and which of the flowers are poisonous when you are camping.

So our goal is to build a portable flower recognition device which enables any user identify flower species easily with a camera and a display screen.

This device will be built with under a $100 budget. Users just need to take a picture of the flower and this device will tell them which kind and what name and attributes of this flower. The uniqueness of this device is that it can work without any internet with a portable size. And the finished product will have a simple UI, and not require a lot of training of users, so it will be a really user-friendly product and is suitable for any ages.

1.2 Background

With Moore's law, the number of transistors in a dense integrated circuit doubles every two years. More and more delicate chips become cheaper than dirt and have great capability of running complex machine algorithms. A cheap embedded computer like Raspberry Pi 3 Model B is capable of implementing real-time vision inference on the basis of complex DNN models.[1] We come up with the idea to run CNN on a cheap and tiny computing device.

Unlike methods using hand-crafted features, our method uses CNN as feature extractor.

In recent years, deep CNN has improved image classification dramatically. Deep learning excels in recognizing objects in images as its implemented using multiple layers of artificial neural networks where each layer is responsible for extracting one or more feature of the image. Combing the advantage of CNN with sufficient refined image data, we can have a flower expert in our pocket.

1.3 High-Level Requirements

The first challenge would be the complexity of CNN on a STM32 microcontroller because something would happen at the very late point of this project is, we have all the elements assembled but it takes forever for the controller to process a machine learning request. Therefore we decide to have the whole design ready through Raspberry Pi first to make sure it could run in a relatively more advanced processor, then move on to STM32.

The second challenge would be the signal connection between all different elements of our design. We need to make sure the communication between camera and the processor, between processor to the display.

The last but not the least challenge would be the soldering of PCB board. The most difficult part of soldering on PCB board would be the STM32 or BCM2873 chip which require point-to-point soldering technique. We don't have any experience of the point-to-point soldering technique, so maybe we will try multiple of times.

2 Design

There are three major parts of our design which may seems pretty simple. However the communication of signals may cause a lot difficulties.

2.0 Software Design

2.0.1 Data Preparation

For this task we will use the data package Oxford FLOWER28/FLOWER102[2] as our base data set and combine it with our self labeled flower images. The Oxford Flower Datasets is created by Oxford Visual Geometry Group by collecting images from various websites. We decide to add some photograph supplementary in order to add some light, pose and scale verifications to the dataset and boost up our accuracy.

2.0.2 Model Selection

Writing and training a deep neural network from scratch requires unaffordable amount of work and data. Thus, transfer learning, using a pre-trained model at the starting point and fine-tune it with target dataset is a very popular approach in doing computer vision and natural language processing tasks. In this project, we are going to utilize the feature extraction layers of pre-trained model and fine-tune the classification layers to get our final flower recognition model.

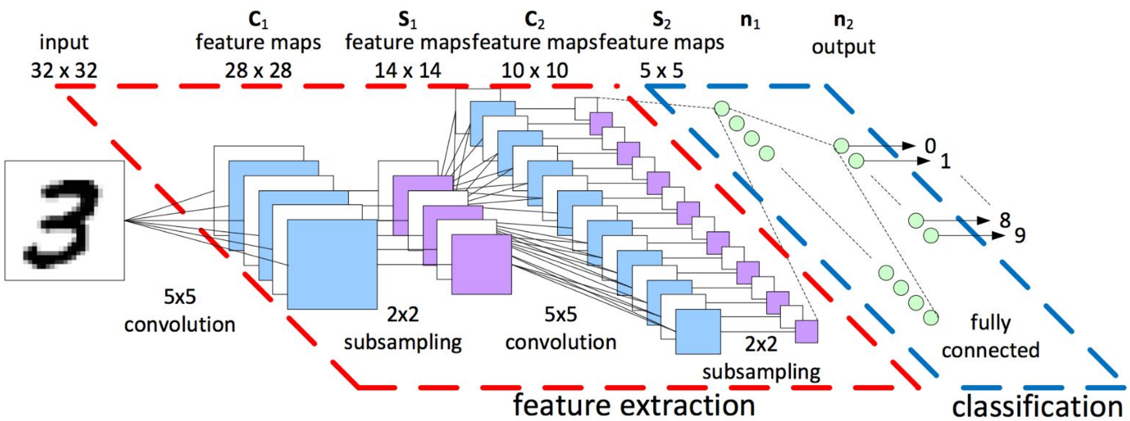

Figure.1 A Convolutional Neural Network that shows different types of layers

Selecting a pre-trained model that fits our goal is the first and the most important step. Since we want to run it on our self-designed hardware, we have to consider the complexity of the model and the lost of accuracy for simpler model architecture. We take the performance of BCM2837 on different models as a reference in Figure 1. It is apparent that the hardware puts a significant limitation on models that requires huge amount of computation. We declined those models with too many parameters and decide to do more research on MobileNet and GoogleNet. We will compare the runtime and accuracy then make our final decision.

Model	Prediction Time in seconds	Graph Load Time in seconds	mAP	Training Steps
ssd_mobilenet_v1	0.72	44.06	0.76	9200
ssd_inception_v2	1.76	50.18	0.78	12000
faster_rcnn_inception_v2	9.89	78.9	0.8	20000
faster_rcnn_resnet50_lowproposals	15.9	80.83	0.82	20000
mask_rcnn_inception_v2	19.4	85.96	None	None
faster_rcnn_resnet50	34.52	87.55	0.82	20000
faster_rcnn_resnet101_coco	None	None	0.83	15000

Figure.2 Benchmarks for different Object Detection Models on Raspberry Pi

2.1 Block Diagram & Physical Design

Figure 3 below shows the connection between each part. There are four major parts based on our original design: control system, power, sensors and user interface. For the power part, we decided to use direct current (cell battery) if the final creation could be fit on a relatively small case, so we could make our device holdable or even wearable. On the other hand, if our device turns out to be too large, we will switch back to normal wall outlet with AC/DC converter.

Ideally, our use interface LCD display would be able to display the image of the flower immediately after capturing it with the camera. If not, it will still meet the minimum requirement of displaying the name of the flower after processing.

After researching cameras online within a reasonable price (~$50), we found the majority of them are able to take photos with high resolutions(at least 640x480) exceeding the input requirement for CNN (224x224x3). Therefore, we believe we can get acceptable photos under limited budget.

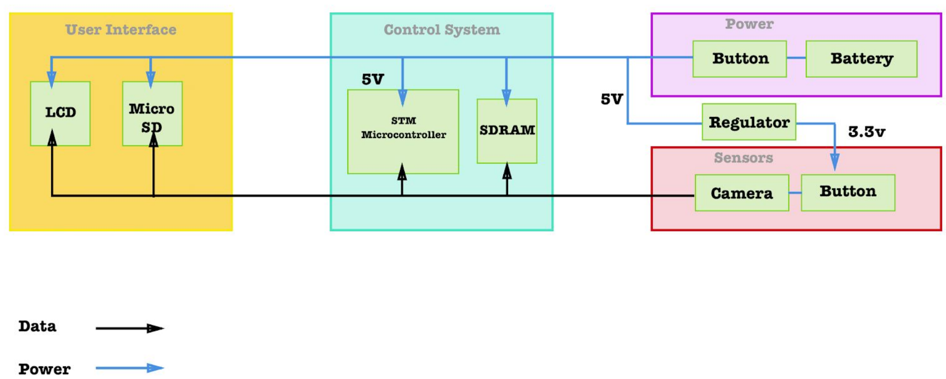

Figure.3 Block Diagram

2.2 Functional Overview

2.2.1 Control Unit

Microcontroller

The microcontroller will read data from the camera as base64 for processing. The DNN model is trained on a regular PC to reduce running time, so our microcontroller will then run already trained algorithm that compares our image with some famous flower dataset such as FLOWERS28 to get a label predicted.

2.2.2 Memory Unit

SDRAM

The SDRAM will be the memory unit for the microcontroller instruction in order to perform the DNN/CNN computation. And it can store the training image if we needed.

SD Card

The SD Card will be used as the transportation device for linux system and image for training and test. So we can update the DNN/CNN model easily and more convenient.

2.2.3 Power Unit

Battery

Our device is powered by a battery set to supply stable 5V voltage to the microcontroller. There could be multiple batteries in one set, so we may need a cell battery holder to sustain our power supply.

Regulator

Some elements require voltages different from 5V, for example, the camera needs a 3.3V supply. Therefore we would need at least one regulator to convert the voltage.

Buttons

There will be at least two buttons for our design. One button is for the overall power supply for both safety reason and general control. The other one would be specifically for the camera to save power.

[Possible] AC to DC converter

If the size of our final device exceeds our expectation, we will consider wall outlet with a 110V AC source. The converter will then supply different voltages to different parts according to requirements.

2.2.4 Sensor Unit

Camera

A normal camera is used as our visual sensor to capture images of live flowers. After taking the picture, the data is sent to the microprocessor as base64 for further analysis.

Button

There will be a button for shutter controlled by users.

2.2.5 User Interface

LCD Display

The image from our camera will be displayed here, and when the microprocessor has done DNN, the trained result would be displayed as the recognized flower type.

2.3 Block-level Requirement

Control unit	**Microcontroller**	Requirement 1: The microcontroller should be able to run the trained DNN/CNN model under linux environment. Requirement 2: The microcontroller should be able to compress the image received from camera. Requirement 3: The microcontroller should be able to support the LCD screen display.
Memory Unit	**SDRAM**	Requirement: The SDRAM should be able to store the instruction from microcontroller in order to perform the DNN/CNN computation.
	SD Card	Requirement: The SD Card should be compatible with the microcontroller and be easily accessed by users.
Power Unit	**Battery**	Requirement 1: It should be stable for power supply for the whole system. Requirement 2: It should provide decent amount of time for power supply since it is a portable device.
	Regulator	Requirement: It should convert the voltage supply from the battery to different parts of board with suitable voltage value.
	Buttons	Requirement : Buttons should accurately control the on and off of the device.

Sensor Unit	**Camera**	Requirement 1: The camera should capture the image with suitable resolution value for the DNN/CNN to run the forward propagation. Requirement 2: The camera should have low amount of energy cost.
	Buttons	Requirement 2: Buttons on this unit will control the shutter for the camera.
User Interface	**LCD Display**	Requirement 1: Display the view when taking the picture as a camera viewfinder. Requirement 2: Display the result information to user after processed by the microcontroller.

2.4 Risk Analysis

The accuracy of our recognition device is our primary measure of success. It is a challenging task to train a good flower recognizer due to the non-rigid deformation, unstable illumination, inter-species similarity and size variation. While using a normal personal computer, the accuracy rate of properly fine-tuned CNN is around 70%~80% depending on the model choice and fine-tune techniques. However, applications demanding both energy efficiency and high throughput must sacrifice accuracy by selecting simpler models or by tuning existing ones[1] since we have to allocate it on our self-integrated platform to collect the data, do the computation and give the result, the accuracy rate could be hard to predict due to the limitation of hardware. The PCB board might require multiple layers, so that would be the risk of the hardware part. If our design requires many layers of PCB board, there will be a risk that we can't assemble the board properly.

3 Safety and Ethics

As our final finished device has the possibility to be put into production in the future, we realize it's really important to understand the ethics and take responsibility for our products. By referring to IEEE Code of Ethics, we as group 14, would like to commit ourselves to the following specific codes:

According to the code #1， #2 and #9 from IEEE Code of Ethics. We should ensure the safety and the welfare to our users and public. We will make sure all the components in our device is properly installed and reduce the potential risk to minimum. If there is unforeseeable accident, we will follow the safety protocol provided by the ECE department and IEEE Electric Committee.

We will hold ourselves with high standard of moral value and integrity. We will give the correct training and explanation to users and public about our product and technology. So, we will strictly follow the code #4 and #5 in order to achieve this goal.

There would be a lot modifications before we reach our goal in the end. Therefore, we would strictly follow #6 and #7 to open to any kind of criticisms to improve our final design. We could also follow code #3 to be honest with our training/testing, and our accuracy rate.

We are also stay respectful to our peers, TAs and instructors regardless of any identities under the guidance of code #8. During our process of designing, creating and adjusting, we would make sure every member of our team can get adequate training on different aspects with the help of code #10.

The safety of our peers and instructors would be our top priority for this project. Since our project will use power and electricity, we would strictly follow operation guidances of each elements to lower any possible risks. There are multiple buttons added to our design to make sure every power could be shut down in case of emergency. We would also follow every rule of the lab to maintain a safe environment for everyone.

4 References

[1] Velasco-Monteroa, D. , Fernandez-Bernia, J. ,Carmona-Gal´ana, R. , Rodrıguez-Vazquez, A. 'Performance Analysis of Real-Time DNN Inference on Raspberry Pi', 2018. [Online]. Available:http://digital.csic.es/bitstream/10261/163973/1/Performance_Analysis_of_Real_Time_DNN_on_RPi.pdf
[2] Visual Geometry Group, Department of Engineering Science, University of Oxford, 'Flower Datasets [Online]. Available:http://www.robots.ox.ac.uk/~vgg/data/flowers/
[3]IEEE, IEEE Code of Ethics Section 7 - Professional Activities. [Online] Available at: http://www.ieee.org/about/corporate/governance/p7-8.html