

---

# Beverage Recognition System SPECIFICATION

for

Final Project

Version 1.0 approved

Prepared by Group4

June 21, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Intended Audience and Reading Suggestions . . . . .	3
1.3	Project Scope . . . . .	3
<b>2</b>	<b>Overall Description</b>	<b>4</b>
2.1	Product Perspective . . . . .	4
2.2	Product Functions . . . . .	4
2.3	User Classes and Characteristics . . . . .	4
2.4	Operating Environment . . . . .	5
2.5	Design and Implementation Constraints . . . . .	5
2.6	Assumptions and Dependencies . . . . .	5
<b>3</b>	<b>External Interface Requirements</b>	<b>6</b>
3.1	User Interfaces . . . . .	6
3.2	Hardware Interfaces . . . . .	6
3.3	Software Interfaces . . . . .	6
<b>4</b>	<b>System Features</b>	<b>9</b>
4.1	Description and Priority . . . . .	9
4.2	Stimulus/Response Sequences . . . . .	9
4.3	Functional Requirements . . . . .	10
<b>5</b>	<b>Other Nonfunctional Requirements</b>	<b>11</b>
5.1	Performance Requirements . . . . .	11

# 1 Introduction

## 1.1 Purpose

### 系統目標

- 提供通過圖片即時識別飲料並展示其詳細信息的系統
- 使用者可以通過上傳飲料圖片來了解此款飲料的配方、過敏原、營養成分以及飲料是否獲得認證。

### 系統範圍

- 接收使用者上傳圖片並顯示結果的網頁前端
- 對圖片進行識別並回傳結果的後端

## 1.2 Intended Audience and Reading Suggestions

本需求規格書用於提供給此專案之開發者以及測試者作為參考，包含有系統概述，功能定義說明，環境需求，使用者界面及操作說明，functional requirements以及nonfunctional requirements等

開發者在閱讀系統概述後應先從functional requirements開始著手設計系統 測試者應首先閱讀操作說明後再行測試

## 1.3 Project Scope

系統包含了提供給使用者上傳所要識別的飲料的網頁以及用於識別飲料並回傳飲料資訊的程式，旨在提供使用者對於飲料資訊更加快捷方便的查詢途徑

## 2 Overall Description

### 2.1 Product Perspective

本系統為一套獨立的系統，分為網頁程式與圖像識別程式兩個部分,如Figure 2.1所示



Figure 2.1: system simple diagram

### 2.2 Product Functions

網頁接收到使用者上傳的圖像後，會將圖像傳至識別程式，由識別程式對飲料種類進行識別後，再回傳結果至網頁端，系統之dataflow diagram如Figure 2.2所示

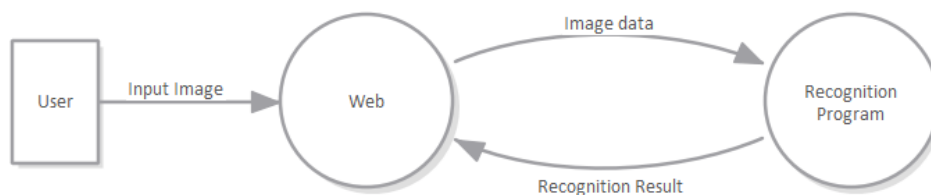


Figure 2.2: dataflow diagram

### 2.3 User Classes and Characteristics

系統主要面向為經常使用手機和平板以及電腦進行上網查詢資料之使用者，潛在使用者對象為較為注重飲食健康，或對某些飲料的配方成分會產生過敏反應的人群

Use case 1	使用 手機/平板電腦 訪問平台之使用者
使用者	進入網頁
系統	網頁顯示使用者介面
使用者	點選”上傳圖片” 按鈕，選擇 拍照/相冊 上傳飲料圖片
系統	網頁顯示飲料之詳細資訊

Use case 2	使用 電腦 訪問平台之使用者
使用者	進入網頁
系統	網頁顯示使用者介面
使用者	點選”上傳圖片” 按鈕，選擇在硬碟中選取文件上傳飲料圖片
系統	網頁顯示飲料之詳細資訊

## 2.4 Operating Environment

服務器端：帶有Unix-like / Windows7及以上作業系統，可正常運作tensorflow程式之設備

使用者：能夠正常瀏覽網頁即可

## 2.5 Design and Implementation Constraints

- 由於飲料品種資料蒐集的時間限制，系統只能支援識別市面上的一部分飲料
- 由於現有技術的限制，系統識別結果無法100%準確

## 2.6 Assumptions and Dependencies

假設

- 使用者上傳並非包含飲料的圖片
- 使用者上傳圖片過程中失敗

外部依賴

- 深度學習套件
- 服務器與使用者需要有正常穩定的網路環境

## 3 External Interface Requirements

### 3.1 User Interfaces

使用者網頁介面需要：

- 1) 可以選擇並且上傳待識別圖片的按鈕
- 2) 用於飲料詳細資訊的區塊

初始介面如Figure 3.1所示，系統需要在使用者上傳圖片，待識別程式識別成功並回傳結果後，在區塊內顯示識別結果之飲料的詳細資訊，如Figure 3.3所示



Figure 3.1: User interface: before uploading image

### 3.2 Hardware Interfaces

系統配置與8G RAM，4 CPUs，160GB固態硬碟之外部主機上

### 3.3 Software Interfaces

- Ubuntu 18.10作業系統
- Python 3.6.8提供Tensorflow運行環境
- PHP7用於架設web server

- 在識別程式中使用OpenCV以三維矩陣的方式讀取外部圖像資料
- 本系統應用Tensorflow之深度學習框架，以及Keras模塊之API

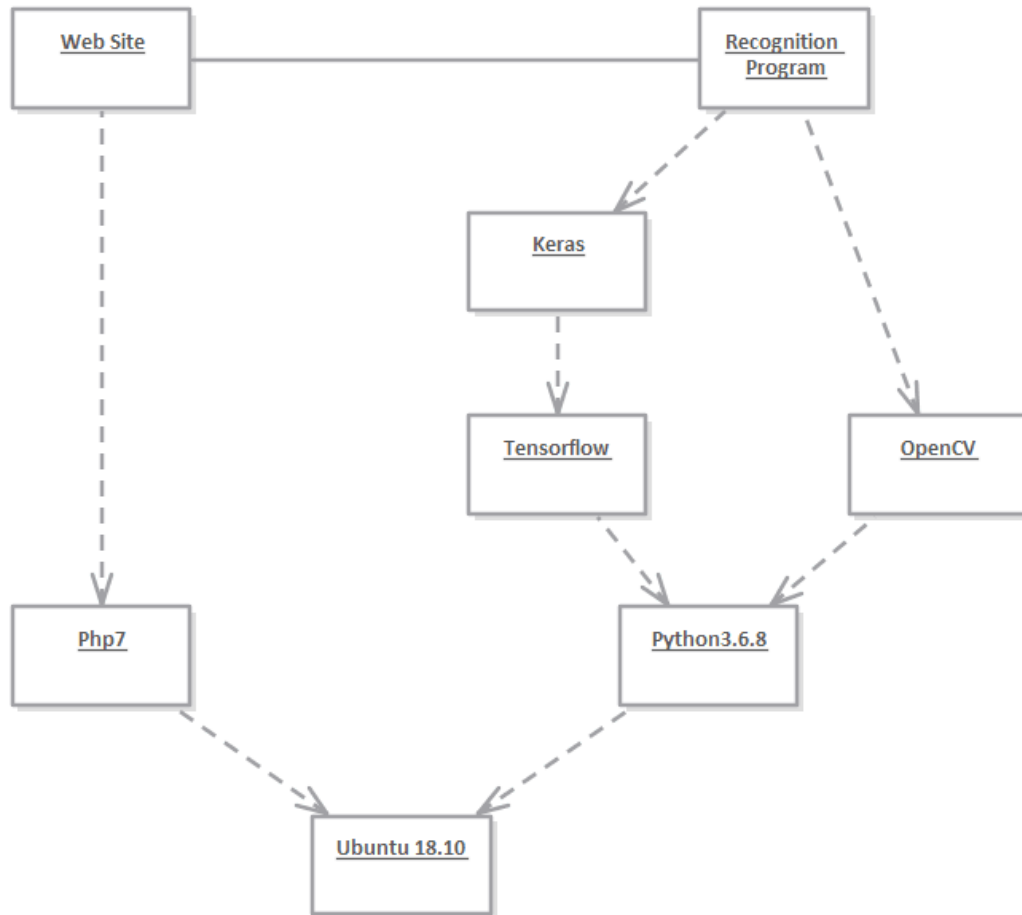


Figure 3.2: Software Interface



Figure 3.3: User interface: show recognition result



## 4 System Features

### 4.1 Description and Priority

由於本系統是用於識別並顯示飲料資訊的系統，所以是否能正確識別出使用者給出的飲料較為重要，所以對於識別的準確度，本系統將放在最高優先級，其次是飲料資訊的詳細程度，其他的輔助項目將視需求會再陸續添加

### 4.2 Stimulus/Response Sequences

Use case 1	
使用者	進入網頁
系統	網頁顯示使用者介面
使用者	點選"上傳圖片"按鈕，上傳飲料圖片
系統	網頁顯示飲料之詳細資訊

Use case 2	
使用者	進入網頁
系統	網頁顯示使用者介面
使用者	點選"上傳圖片"按鈕，上傳模糊不清的飲料圖片
系統	網頁顯示飲料資訊，但識別準確率下降

Use case 3	
使用者	進入網頁
系統	網頁顯示使用者介面
使用者	點選"上傳圖片"按鈕，上傳用難以展示飲料特征之角度拍攝的飲料圖片（如正上方）
系統	網頁顯示飲料資訊，但識別準確率下降

Use case 4	
使用者	進入網頁
系統	網頁顯示使用者介面
使用者	點選"上傳圖片"按鈕，上傳不包含飲料的圖片
系統	網頁顯示錯誤的飲料資訊

### 4.3 Functional Requirements

REQ1	圖片上傳
說明	提供使用者上傳待識別飲料圖片的功能
輸入	由使用者點選功能選項觸發
處理	將所選圖片傳至識別程式
輸出	網頁顯示上傳中之加載圖標

REQ2	飲料識別
說明	提供識別飲料並傳回結果的功能
輸入	識別程式接到使用者上傳的圖片後觸發
處理	識別圖中飲料並回傳識別結果
輸出	飲料種類

REQ3	飲料資訊顯示
說明	提供識別結果所示飲料的詳細資訊
輸入	接到飲料識別程式所傳回識別結果後觸發
處理	讀取識別結果所示飲料的詳細資訊
輸出	網頁顯示識別結果所示飲料之詳細資訊

## 5 Other Nonfunctional Requirements

### 5.1 Performance Requirements

識別準確度：

- 由於本系統是用於提高使用者查詢飲料資訊效率的即時識別系統，所以對於飲料種類的單次識別應具有80%以上的準確率

識別速度：

- 由於本系統旨在提高使用者查詢飲料資訊效率，所以對於飲料種類的單次識別時間應控制在5s以內

可移植性：

- 系統應在Unix-like作業系統與Windows作業系統上進行移植後可以正常運作

可測性：

- 系統提供的功能均可以進行測試，以確保系統運作時的正確性

可維護性：

- 系統提供的功能在建構時應加以模組化，以便後期易於維護以及功能易於增加或縮減

穩定性：

- 本系統應具有較高的穩定性，能在99%的時間內正常運作而不需要重啓系統