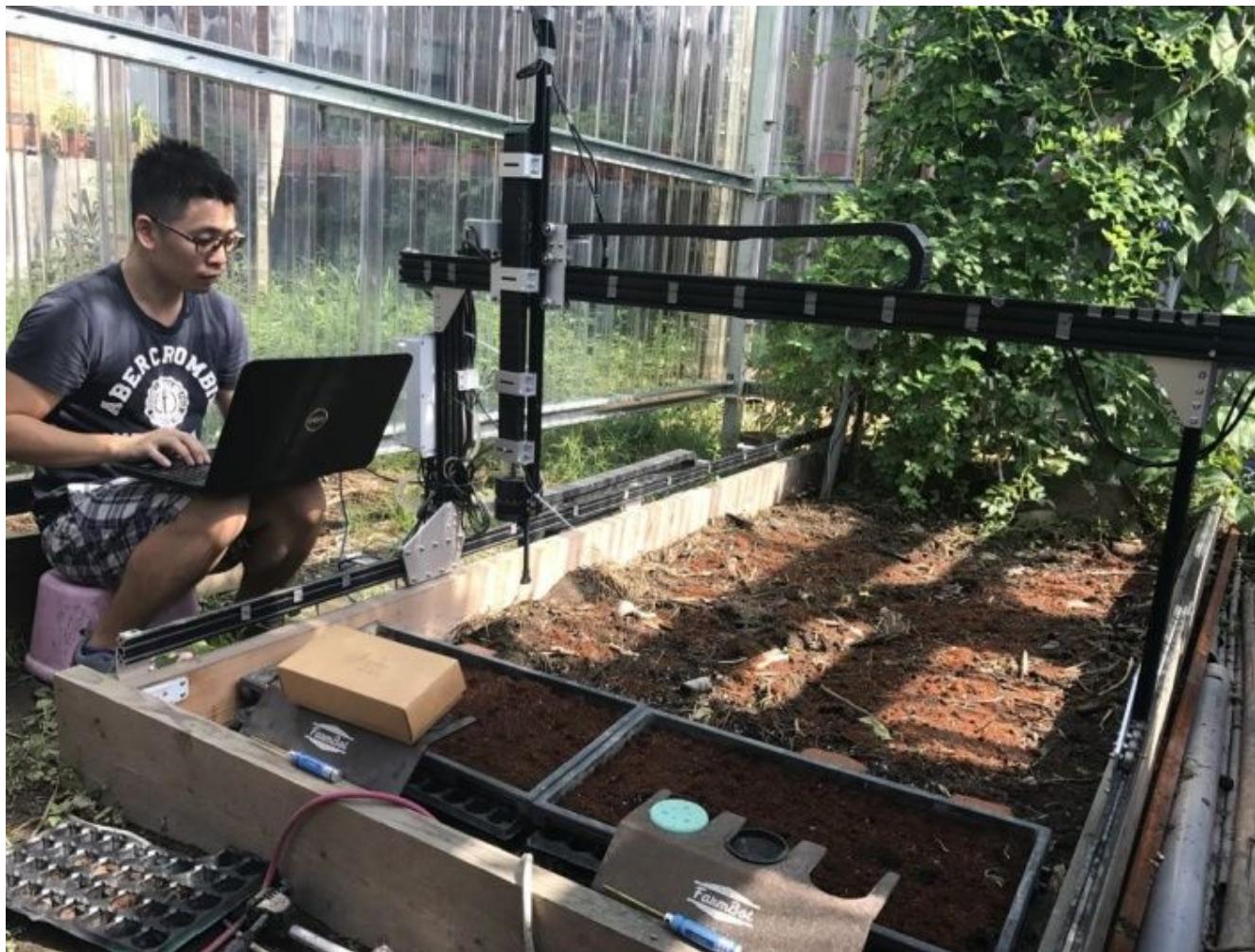


MiniFarmBot智慧農場的UI介面 設計以及OpenCV的應用



Introduction of Speaker

Ying Yao (姚 頬)

Background

- ✓ Mechanical Engineering
- ✓ Measurement Engineer
- ✓ Image Engineer
- ✓ Machine Learning Engineer
- ✓ A Newbie Maker

Experience

- ✓ Generative Model (VAE, GAN, cycle-GAN ...)
- ✓ Fast RCNN, Mask RCNN

Maker Experience

- ✓ 2017/01 Join FBTUG
- ✓ 2017/03 Attending the 1st Farmbot Assembly in Taiwan
- ✓ 2017/11 Join Maker Faire Taipei 2017
- ✓ 2018/03 Talker in FBTUG Annual Meeting



Outline

- Introduction of Farmbot
- What is FBTUG
- Introduction of FBTUG Commander
- Python Practice



Introduction of FarmBot

- Introduction Video of FarmBot Genesis:
<https://www.youtube.com/watch?v=8r0CiLBM1o8&feature=youtu.be>
- FarmBot Website:
<https://farm.bot/>
- Growing Food in Space, on the Moon, and Mars:
<https://forum.farmbot.org/t/farmbot-nasa-collaboration/2255>



Introduction of FarmBot

- **FarmBot**
 - an *open source* precision 3DP structured farming machine, software and documentation including a farming data repository.
 - its targets is "*Create an open and accessible technology aiding everyone to grow food and to grow food for everyone.*".





FarmBot

🔗 <https://farm.bot>

Repositories 4

People 2

Projects 0

Grow your team on GitHub

GitHub is home to over 28 million developers working together. Join them to grow your own development teams, manage permissions, and collaborate on projects.

Sign up

Dismiss

Pinned repositories

Farmbot-Web-App

Setup, customize, and control FarmBot from any device

● TypeScript ⭐ 483 ⚡ 158

farmbot_os

The operating system and all related software that runs on FarmBot's Raspberry Pi.

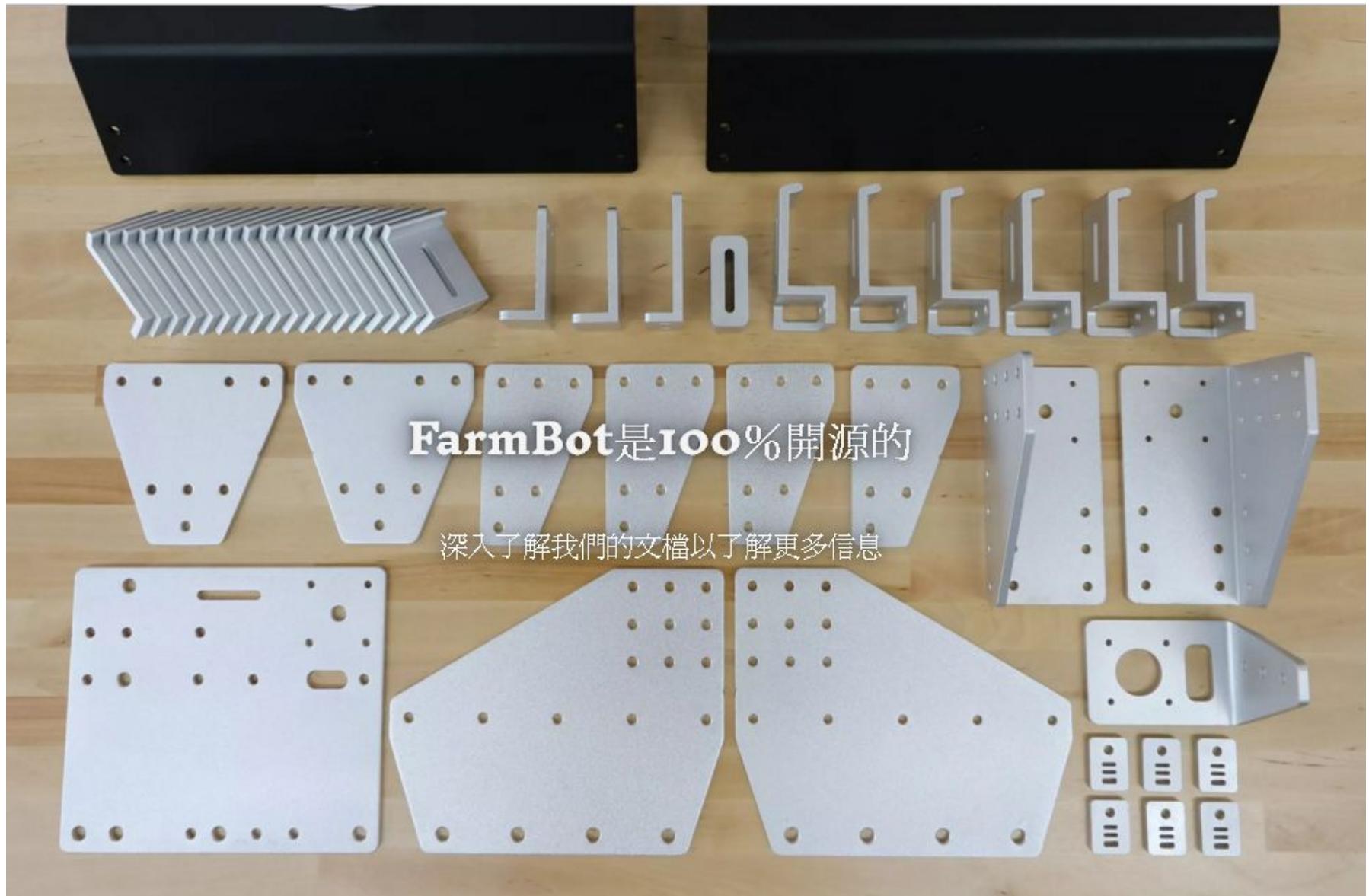
● Elixir ⭐ 496 ⚡ 154

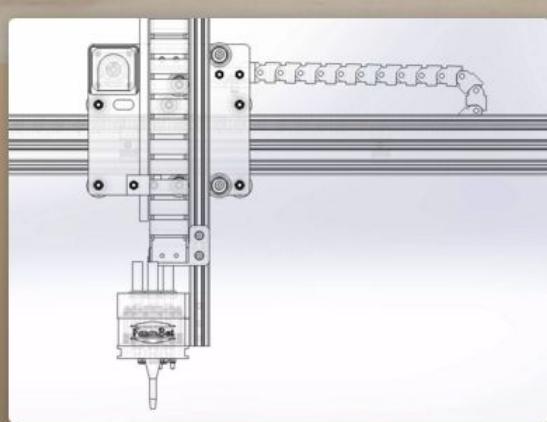
farmbot-arduino-firmware

Arduino firmware that executes g-code like commands over a serial line to move the FarmBot and operate the tool mount

● C++ ⭐ 225 ⚡ 109







高級硬件

我們不遺餘力地設計，製造和採購最高質量的部件。此外，所有CAD模型都是公開的，因此您可以構建自己的零件。

[查看硬件畫廊](#)

```
1 import React from 'react';
2 import { sendCommand } from '../../actions/bot_actions';
3
4 export class DirectionButton extends React.Component {
5   sendCommand() {
6     var payload = { name: "moveRelative", speed: 100 };
7     var isNegative = (this.props.direction == "up") ||
8       (this.props.direction == "right");
9     var multiplier = (isNegative) ? -1 : 1;
10    payload[this.props.axis] = (this.props.steps || 250) *
11      this.props.dispatch(sendCommand(payload));
12  }
13
14  render() {
15    var classes =
16      "button-like fa fa-2x arrow-button radius fa-arrow-"
17      + this.props.direction;
18    return <button onClick={this.sendCommand.bind(this)} className={classes}>
19      <i />
20    </button>
21  }
22}
```

軟件源代碼

幫助我們構建新功能，為您自己的自定義mod分叉代碼，或在本地託管Web應用程序以實現真正的離網獨立性。所有軟件均在MIT許可下。

[在GITHUB上請求我們](#)



文檔

通過我們的逐步裝配說明，物料清單和完整的技術規格，了解FarmBot的內部和外部。文檔屬於CCO下的公共領域。

[閱讀DOCS](#)



可互換工具

標準工具包括種子注射器，澆水噴嘴，土壤傳感器和除草機。

[探索工具](#)



離網

使用小型太陽能和雨水收集系統將FarmBot帶離網。

[看看它做得怎麼樣](#)



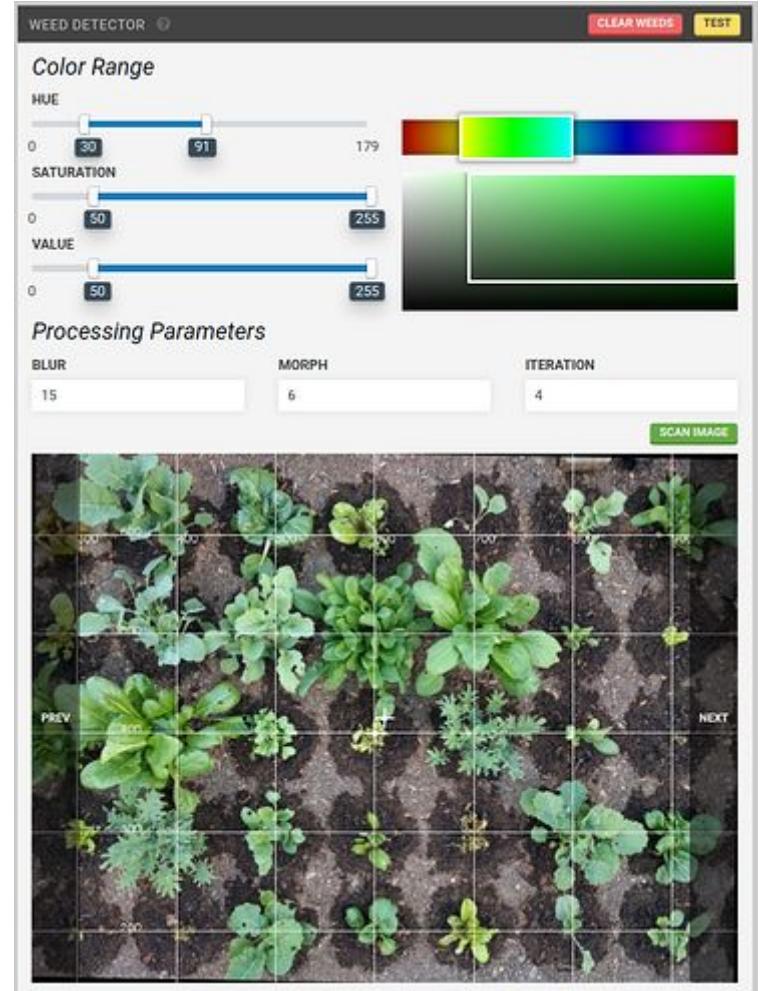
延長生長季節

在溫室中使用FarmBot全年種植新鮮蔬菜。

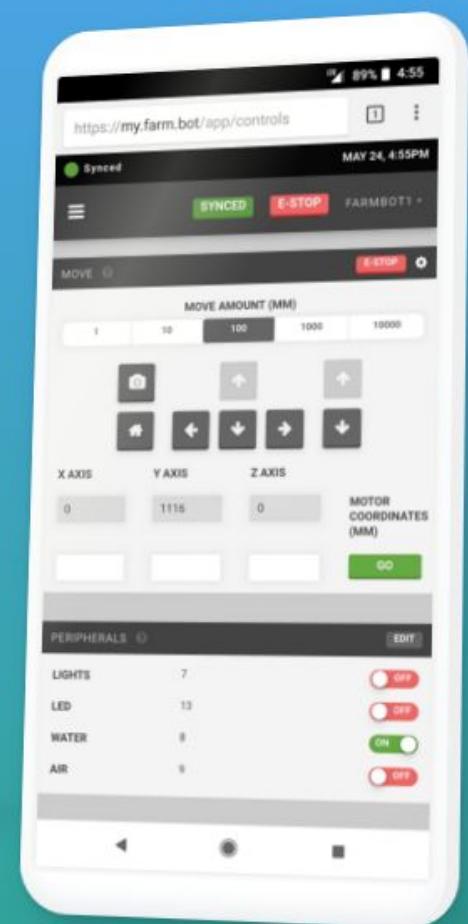
[學到更多](#)

UG

Happy Farm in Real World



從任何地方農場





你將如何使用**FarmBot**？



在教育領域

世界各地的學生，教師，教授和教育機構都在使用FarmBot作為基於STEM的學習和研究工具。

[學到更多](#)



對於糧食主權

在凸起的床，城市屋頂或溫室中使用**FarmBot**為自己，家人和社區種植食物。

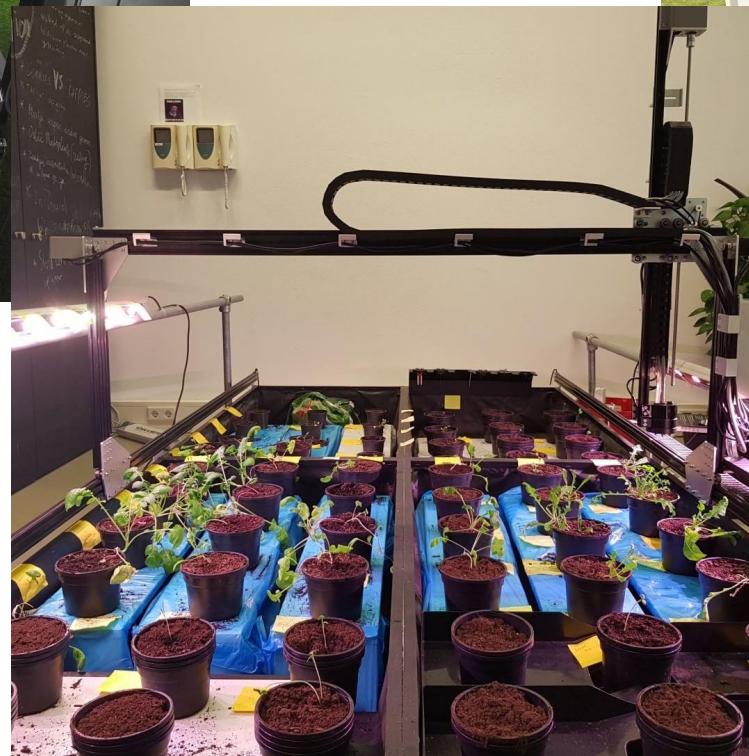
[訂購你的FARMBOT](#)



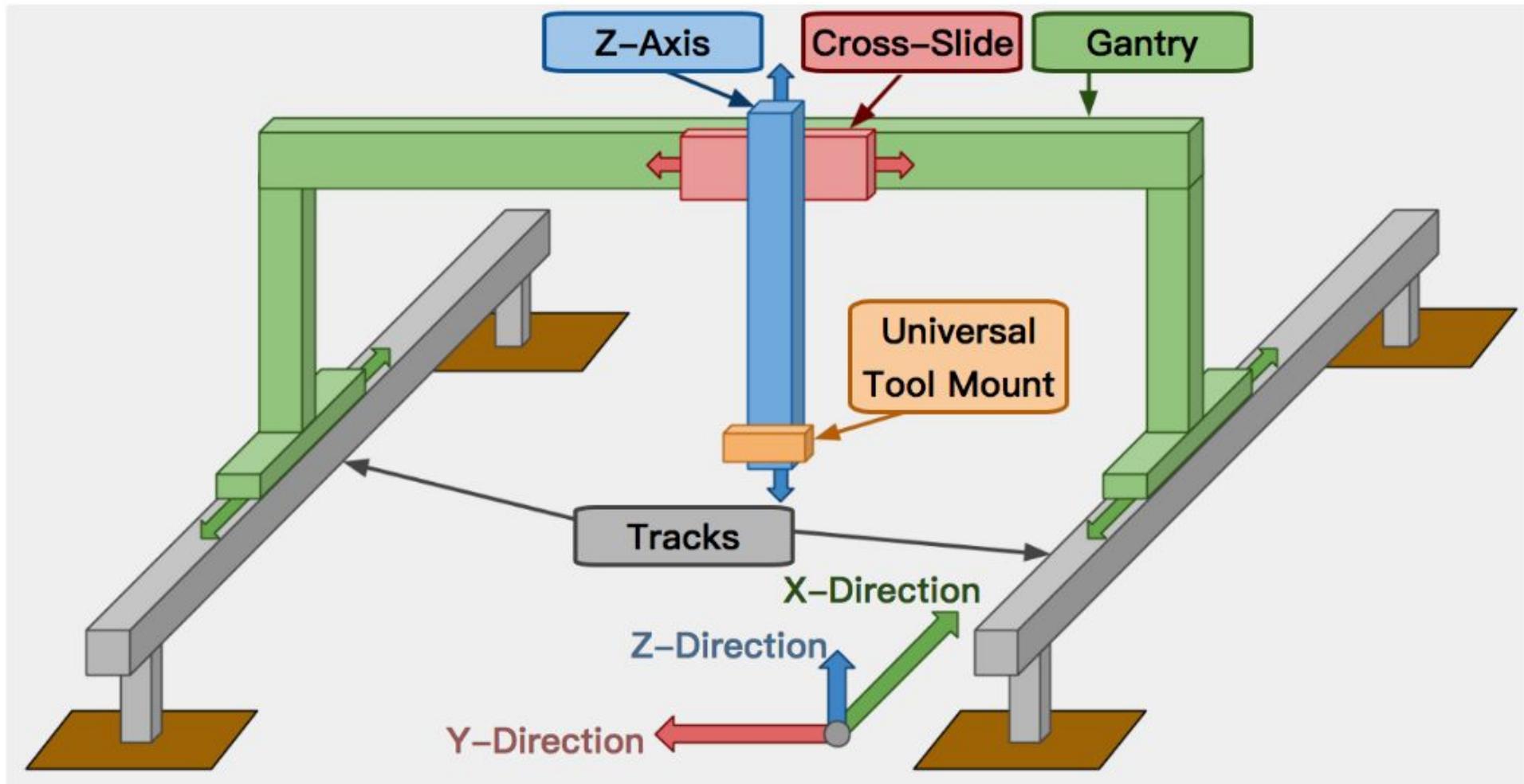
商業

大大小小的農民可以使用**FarmBot**來降低勞動力成本，提高效率，控制投入，並測試新的種植方法。

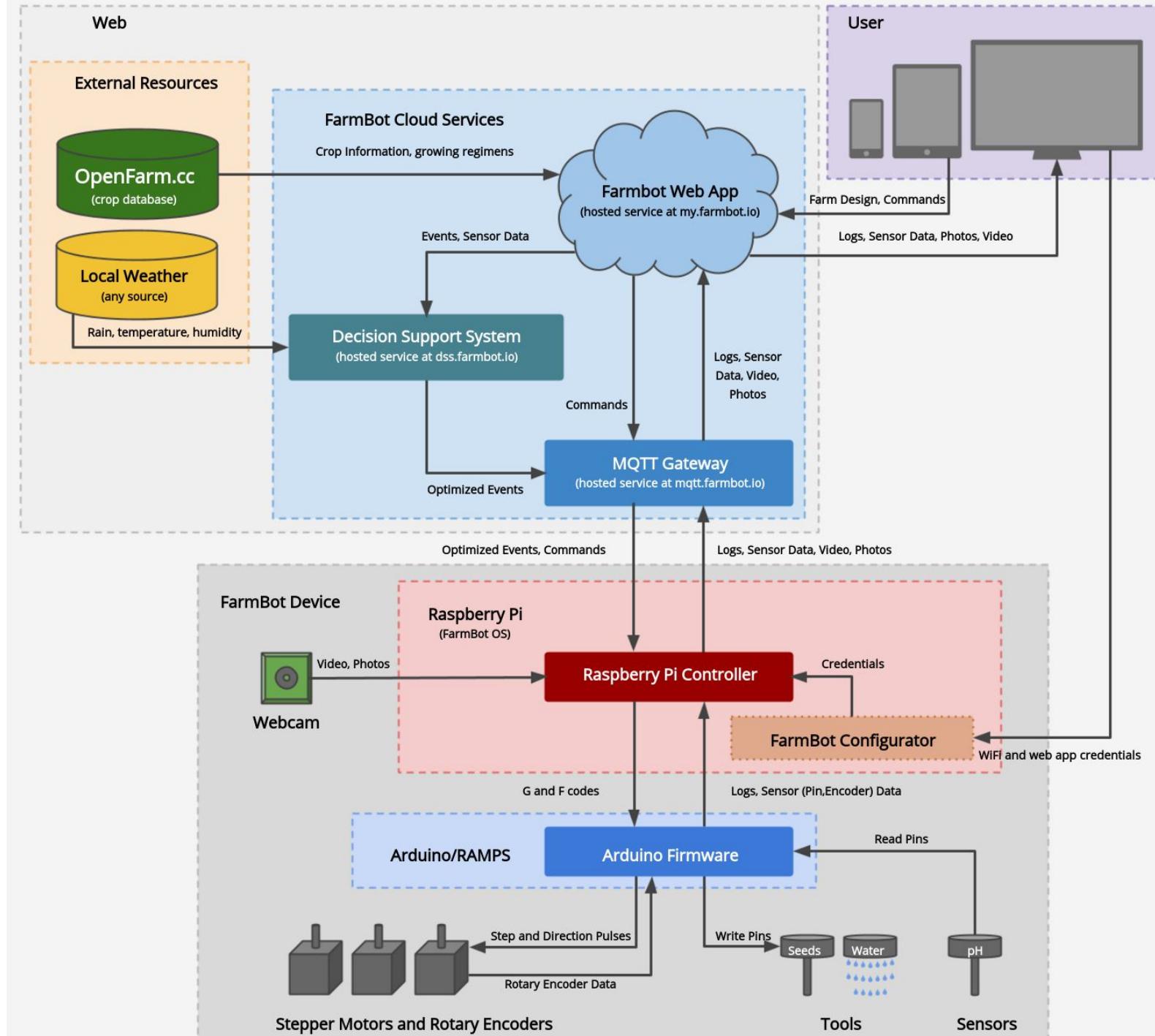
[閱讀更多](#)



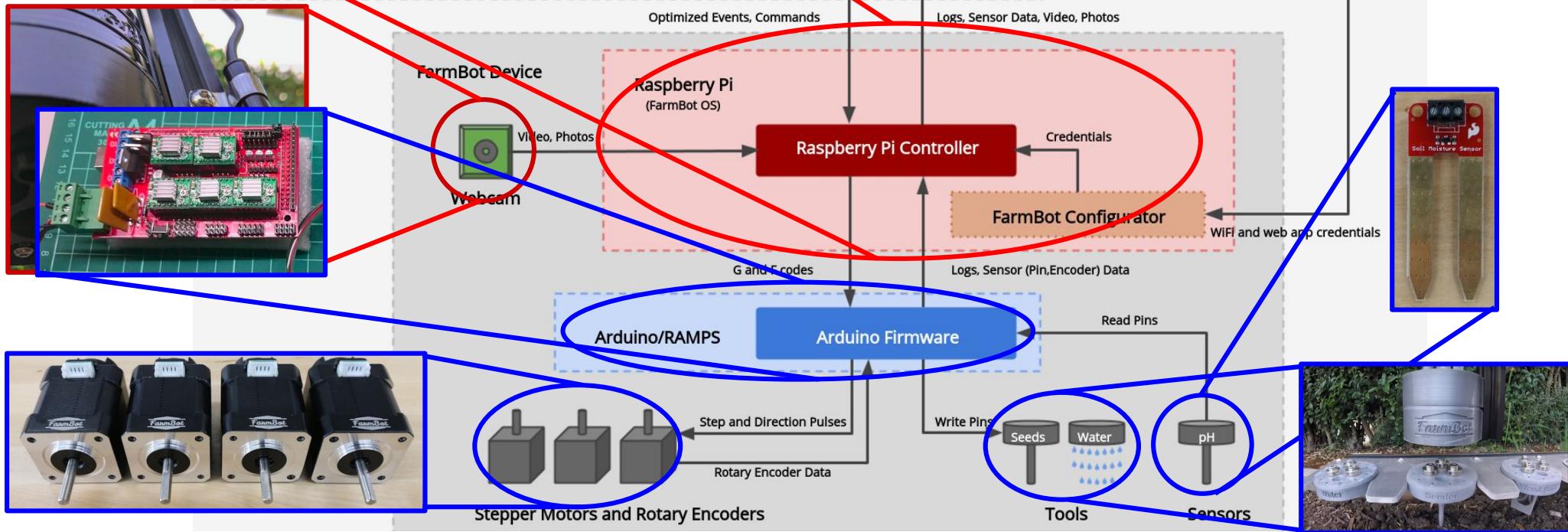
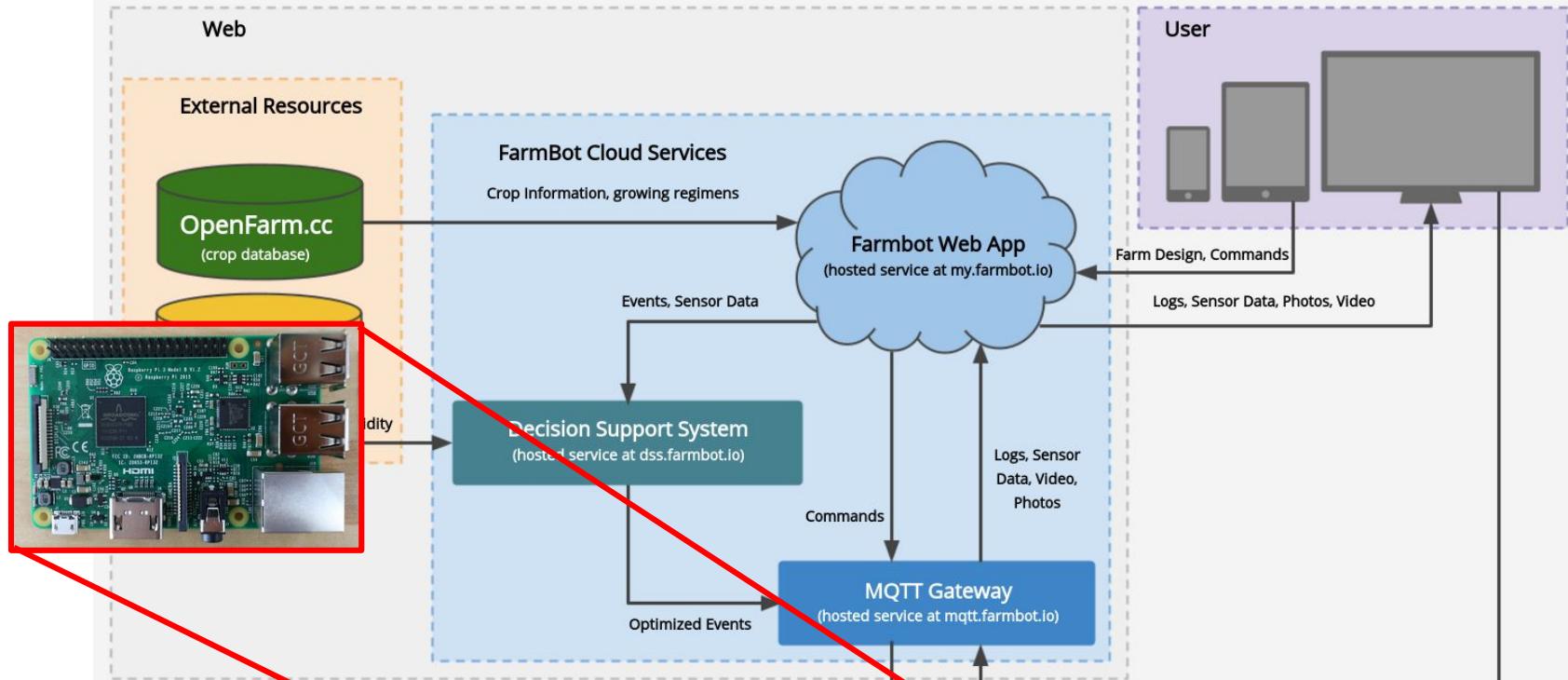
Hardware Structure



FarmBot Software High Level Overview



FarmBot Software High Level Overview



News of Farmbot in Taiwan

【台版FarmBot計畫】打造自動化農場！

<https://makerpro.cc/2017/05/farmbot-taiwan-user-group/>

開源的發展，能大幅降低進入門檻，購買價格也變得非常低廉，使得各類應用能夠蓬勃發展，也會因而促使新的產業誕生。如果我們不盡早做出行動，那麼在世界上某個地方，也會發展出同樣的產業體系，結果我們又會落後，趕不上別人的發展。

目前的 FarmBot發展日趨成熟，其實已經對整個世界做出巨大的貢獻，在未來發展上，能連結更多感測裝置與外接設備，達到完全智能監控、自動採收、除蟲等功能。哈爸期盼台灣的團隊能自行打造出一台獨一無二的FarmBot，並回饋一些理念給原廠，成為國際社群的一份子。

而我們目前也已經舉辦過兩次的開箱與組裝活動，參加的人數十分踴躍，歡迎對Farmbot有興趣的朋友們，也能加入Farmbot Taiwan User Group，一起促進台灣科技農業的成長！

作者：哈爸、楊哲寧



FarmBot Taiwan User Group (FBTUG)

https://www.facebook.com/groups/FarmBotTUG/?ref=br_rs

為何要做：

- 希望藉由開源農業機器的推廣與貢獻，給友善農業一點幫助

誰該加入：

- 對 FarmBot 的設計、製造、使用有興趣
- 想給予全球 FarmBot 社群一些本地貢獻
- 對開源農業機器的設計、製造、使用有興趣
- 對友善農業，科技農業，開源農業有興趣

建議做什麼：

- 分享 FarmBot 研究，設計，製造，使用相關訊息
- 分享 開源農業，友善農業，科技農業相關訊息

PS: 此為自發性社團，並非 FarmBot 正式 User Group

by 哈爸

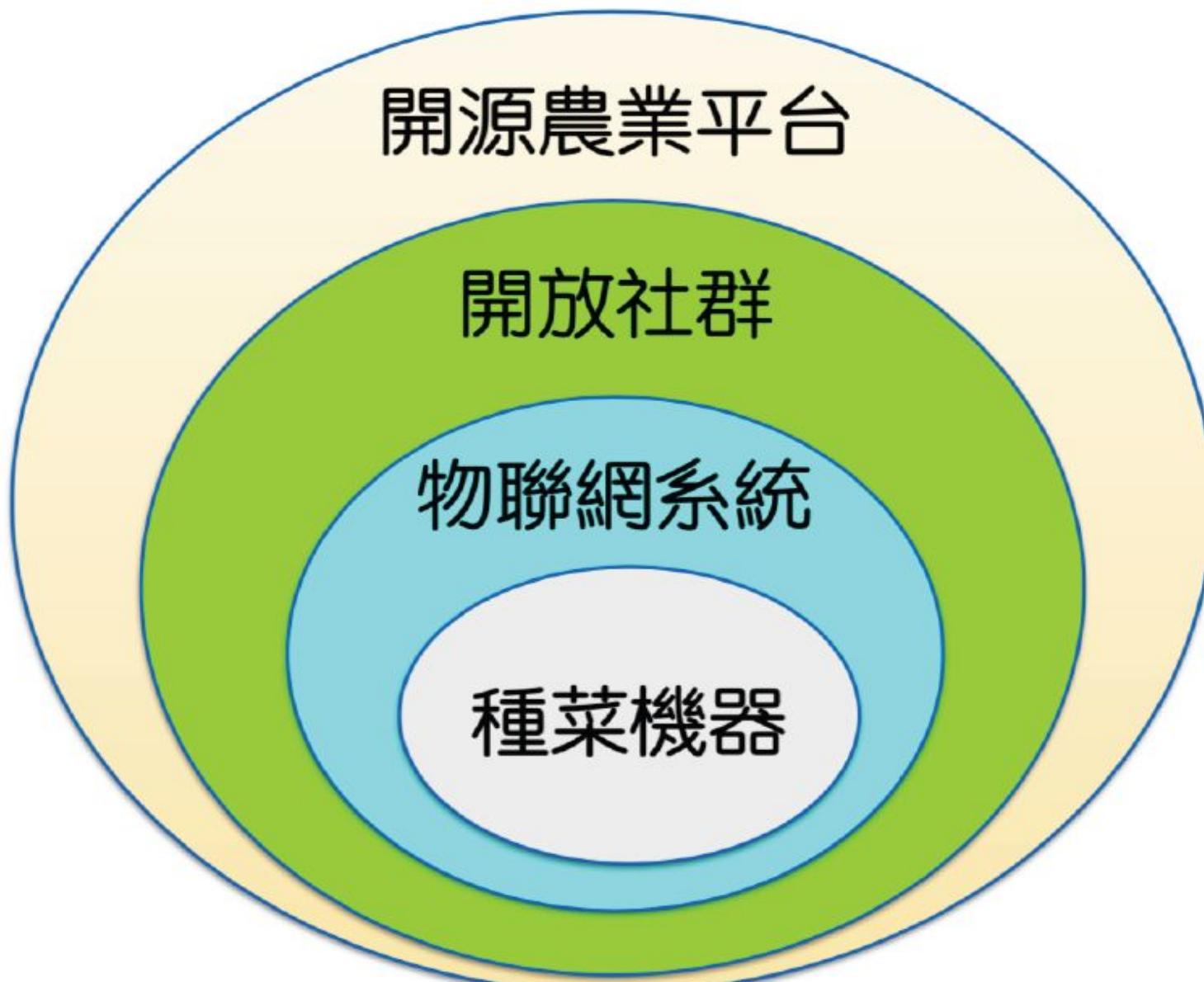


FarmBot Taiwan User Group (FBTUG)

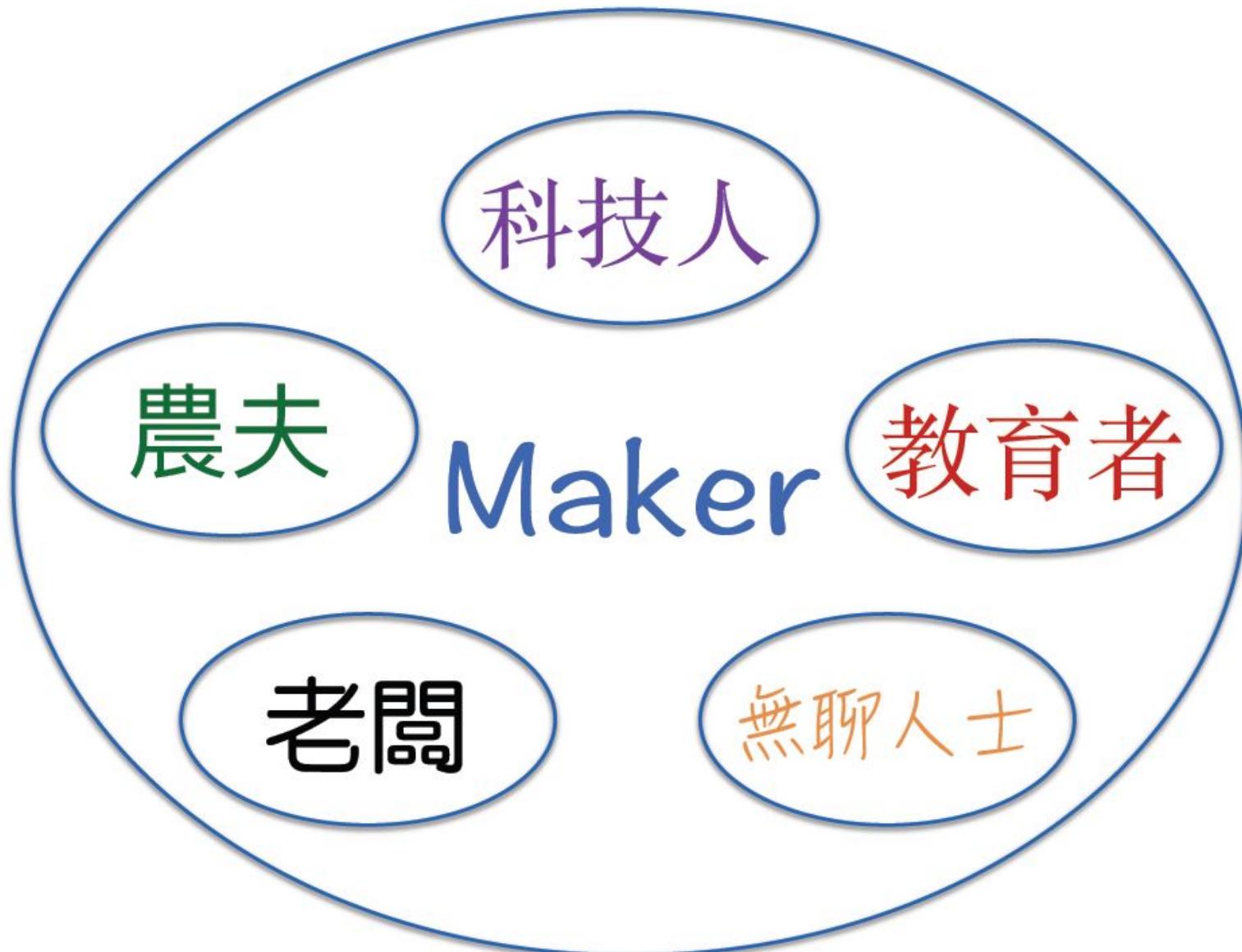
- FBTUG Activities



What is FBTUG



Who joins FBTUG



FBTUG related Documents

- [FBTUG-Introduction slides](#)
- [FBTUG文件首頁](#)
- [Quick Start Guide](#)
- [FBTUG-Q&A and FAQ](#)

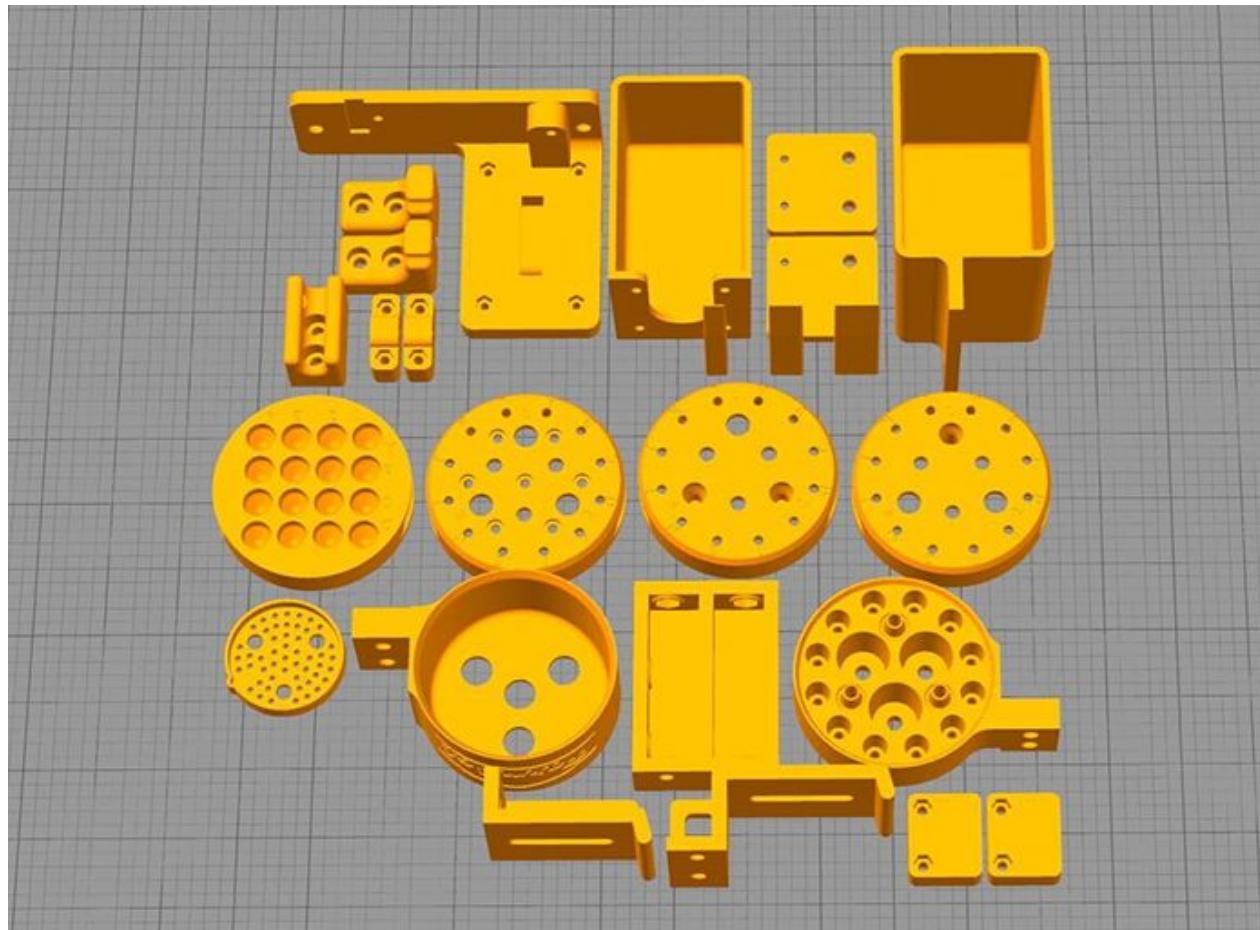


FBTUG Project

- **Revising FarmBot Component:**

https://github.com/FBTUG/FBTUG_MiniFarmBot/tree/master/Hardware/S

TL



FBTUG Project

- **Revising FarmBot Component:**

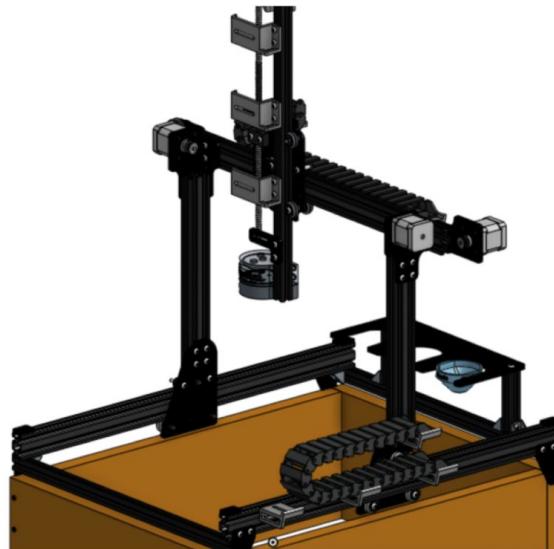
https://github.com/FBTUG/FBTUG_MiniFarmBot/blob/master/Hardware/CAD/MiniFarmbot_Acrylic_plate.dxf



FBTUG Project

- **mini Farmbot:**

- 侯專益(Joe) | FBTUG 的推廣者BIOS 資深部經理
(Insyde) | miniFarmbot 設計者
- MiniFarmBot Demos:
https://www.youtube.com/watch?v=0hh_v_is2nw&index=7&list=PL29zJVsy016_BPGLUsIHnnMMVlxPBNySL



MiniFarmbot 開源分享

(Joe hou) hou.joe@gmail.com

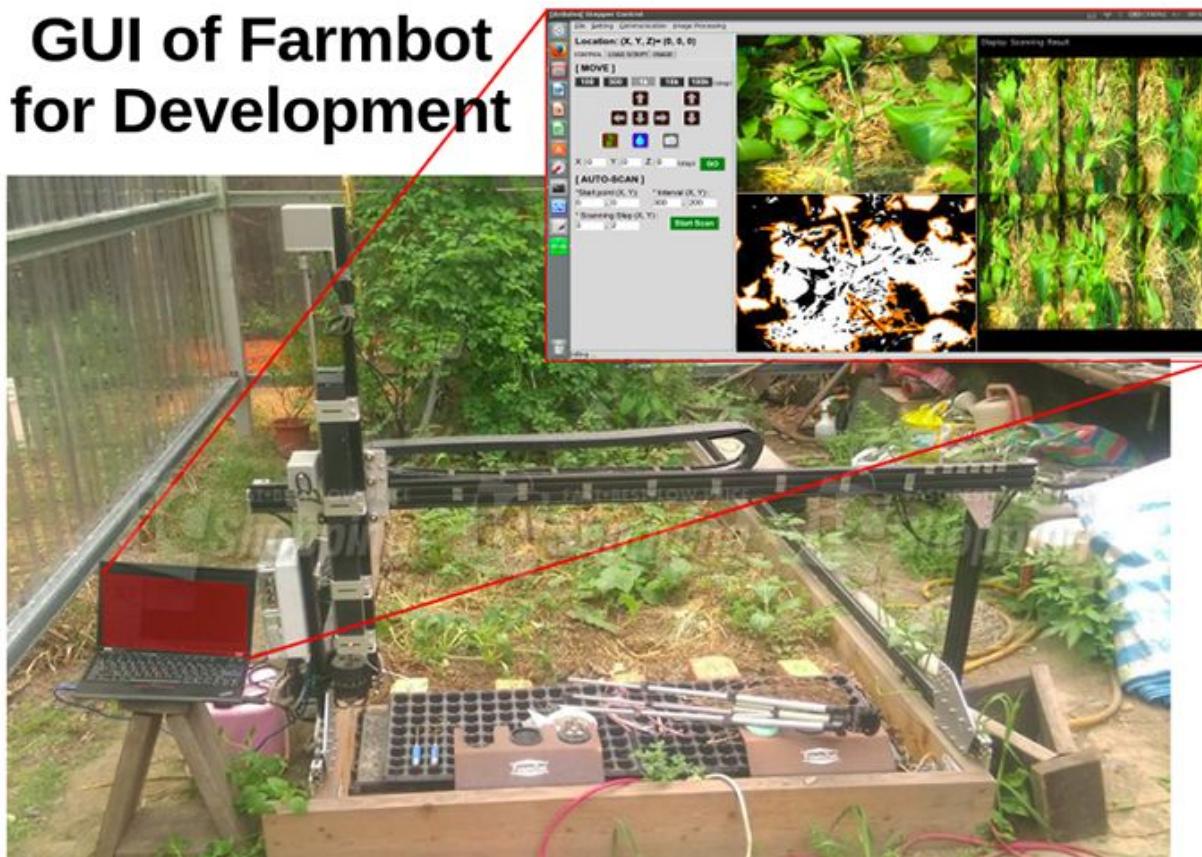


FBTUG Project

- **FBTUG Commander:**

https://github.com/FBTUG/DevZone/tree/master/FBTUG_Commander

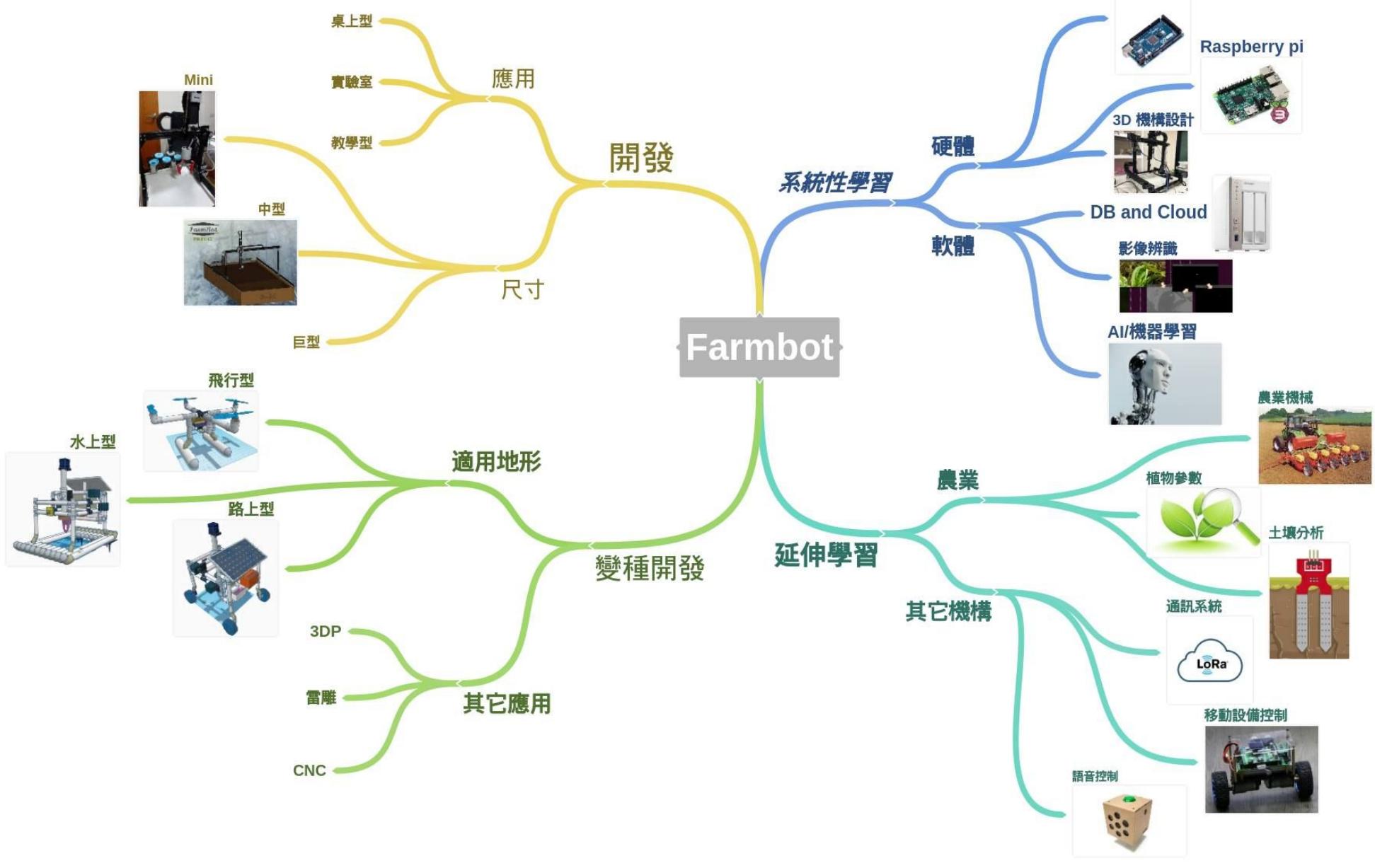
**GUI of Farmbot
for Development**



- Lower barrier to entry for beginner
- Integrate seeding, motor motion, watering and camera control
- Load & Run script to complete series actions
- Auto scanning specific regions

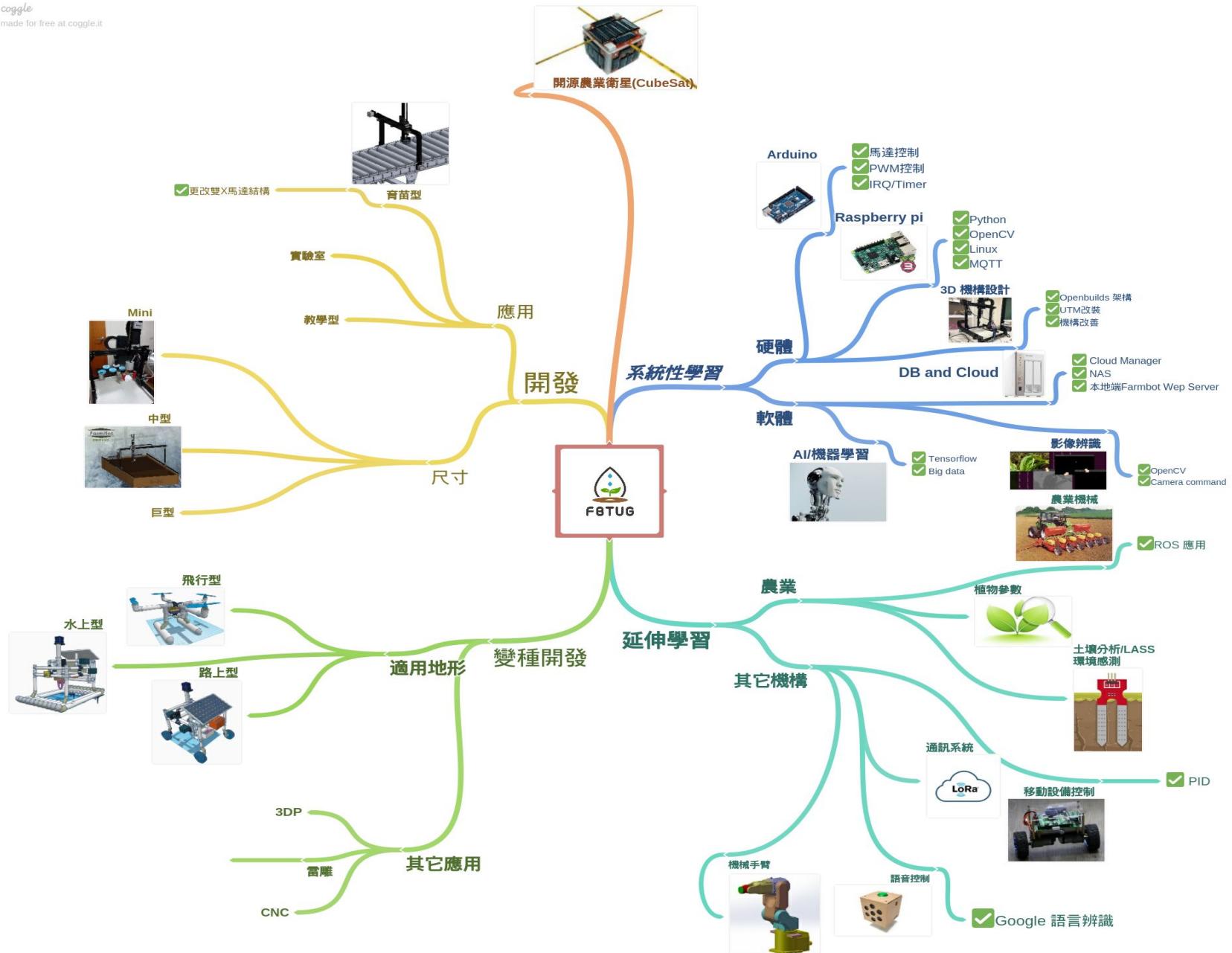
Potential of FarmBot

coggle
made for free at coggle.it



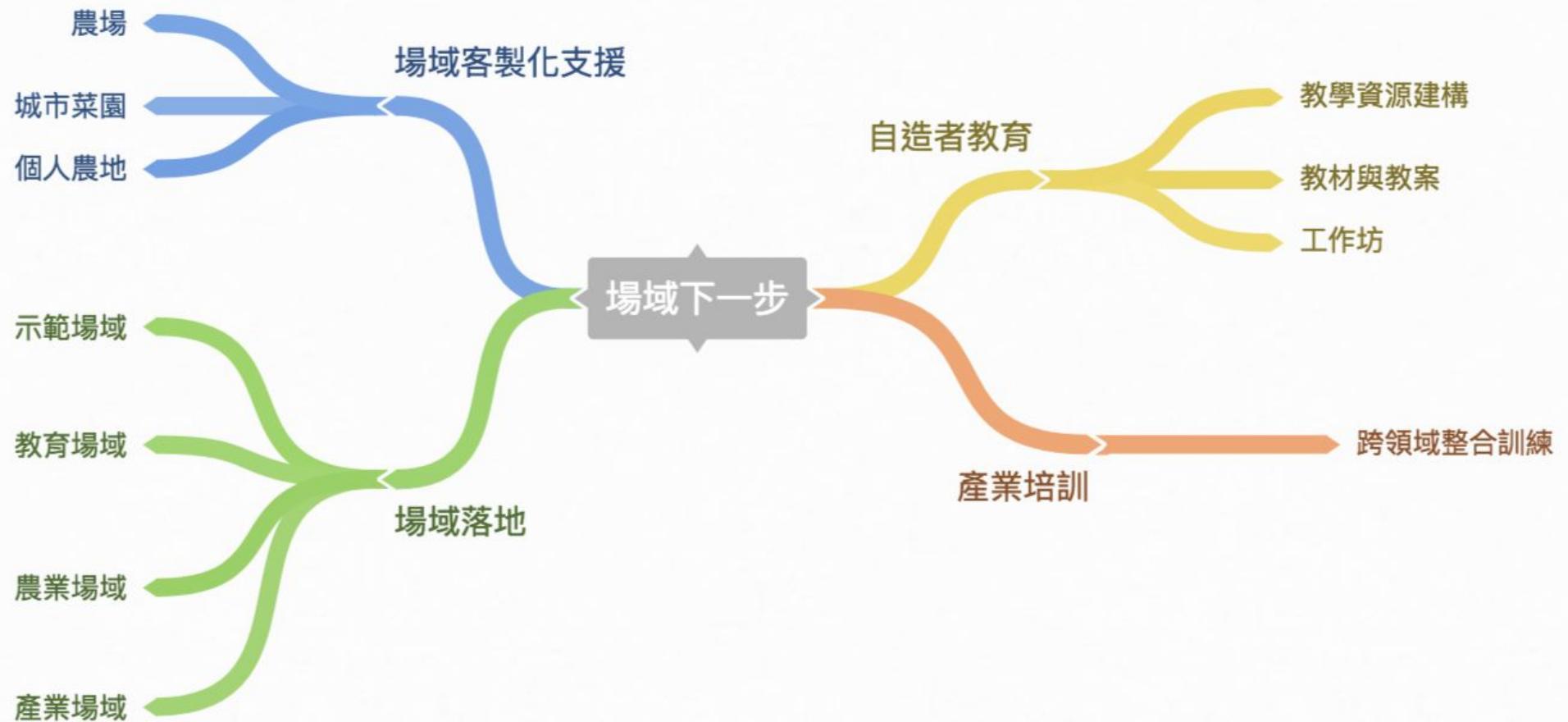
Potential of FBTUG

coggle
made for free at coggle.it

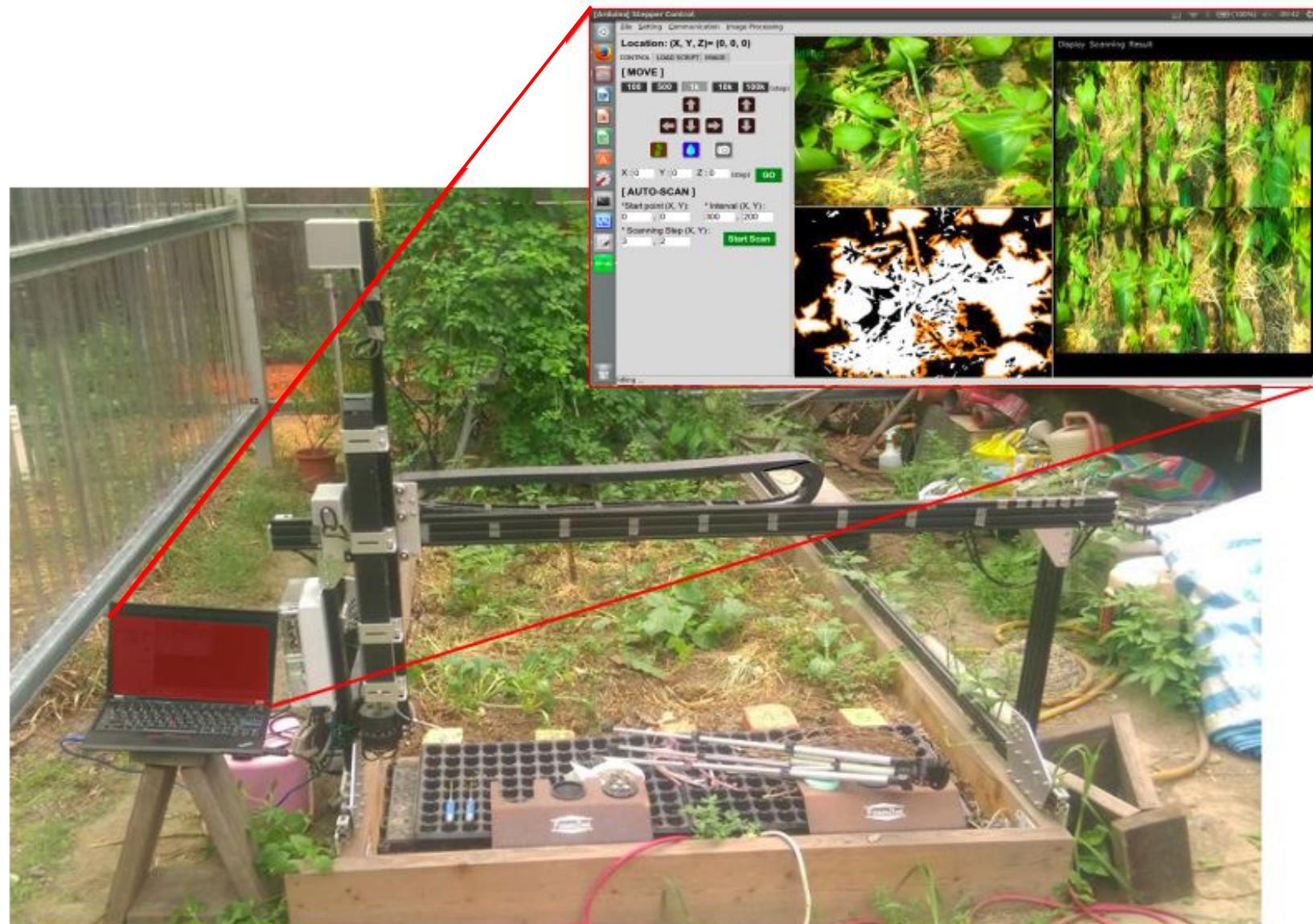


TUG

Potential of FBTUG



Introduction of FBTUG Commander



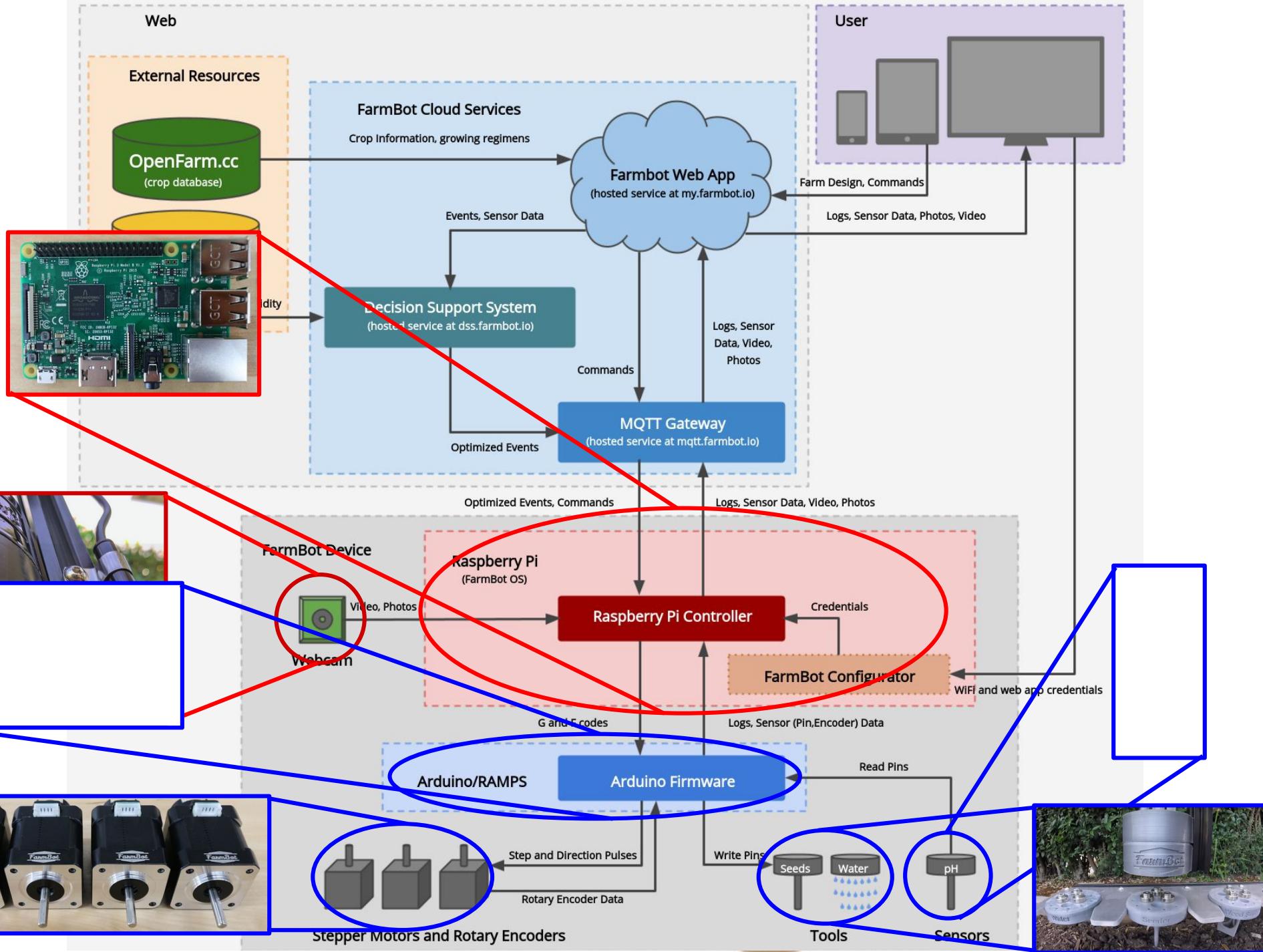
Outline

- Farmbot structure
- Motivation
- Introduction of FBTUG Commander
- Summary



FBTUG Annual Meeting 2018

FarmBot Software High Level Overview

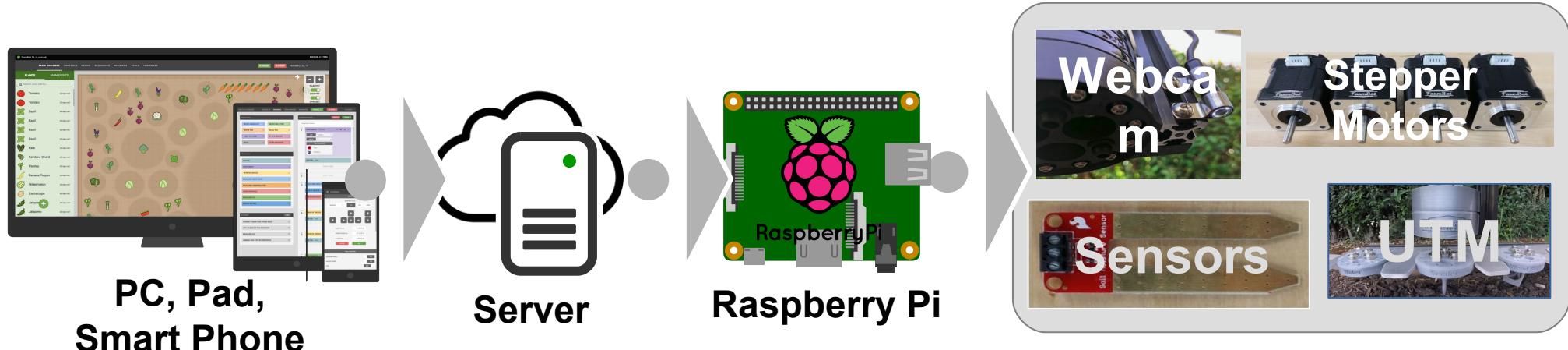


Relating Knowledge behind Farmbot

- **Hardware:** Mechanism, Assembly...
 - Variety of parts and elements
 - Variety of tools
 - Adjustment of hardware
 - Mechanical Design
- **Firmware:** Wiring, Signal, Control...
 - Wiring of signal I/O
 - Circuit logic
 - Arduino Code (C/C++)
- **Software:** OS, Web App, Camera...
 - OS system
 - Web: Ruby on Rails, RESTful JSON API, MQTT
 - Image Process lib: opencv...
 - Machine Learning lib: sklearn, tensorflow, keras...



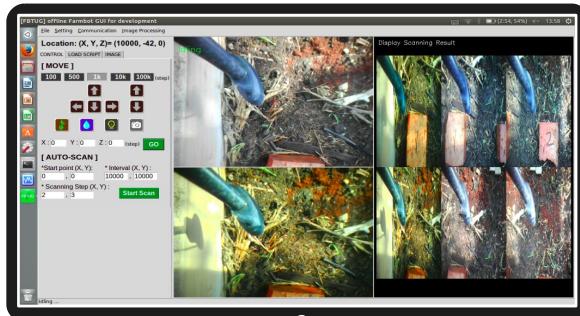
Farmbot Concept Structure



FBTUG Commander Concept Structure



FBTUG Commander



Control



Webcam



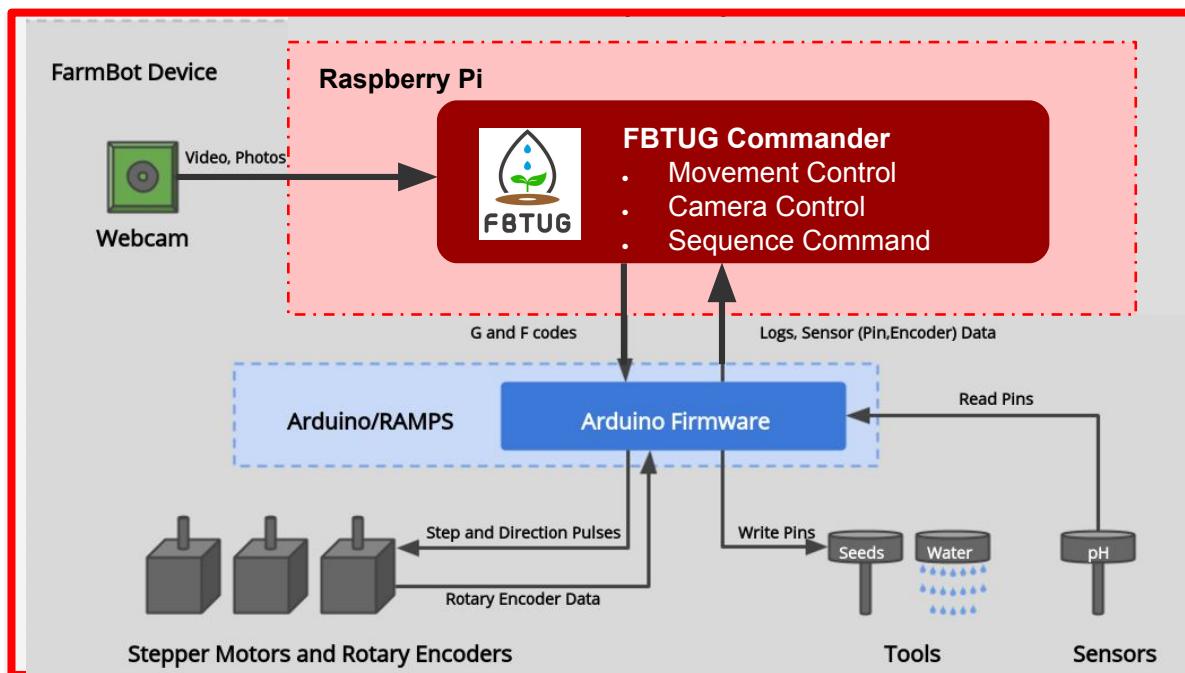
Stepper Motors



UTM



Sensors



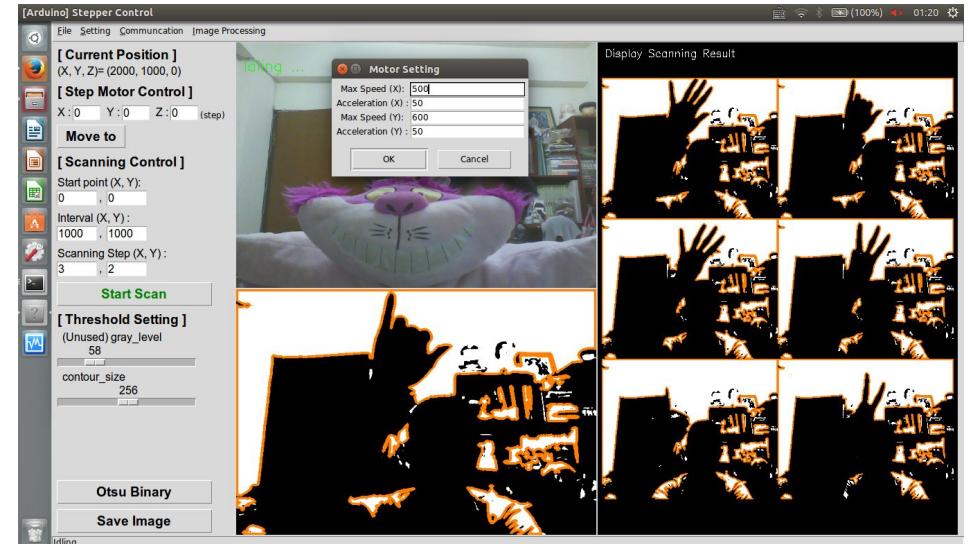
TUG

FBTUG Commander

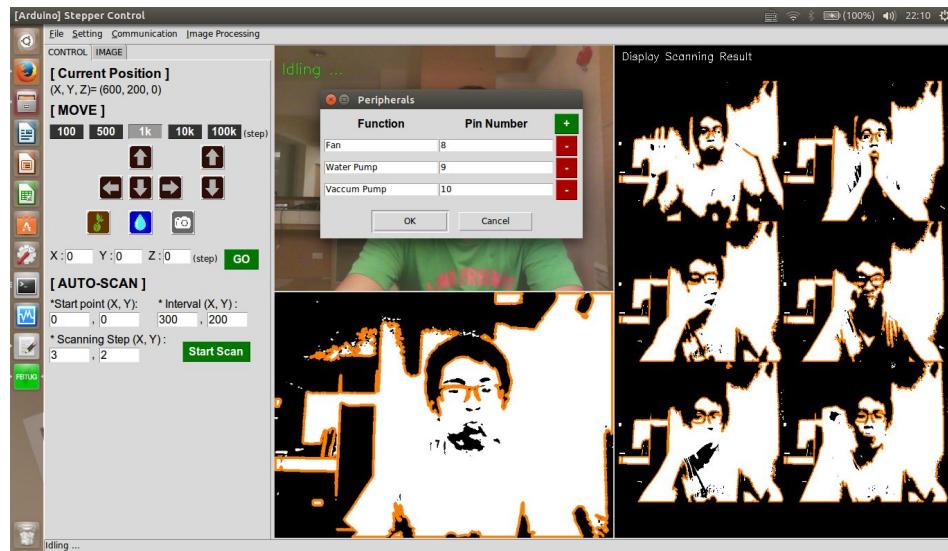
• Ver 0.0



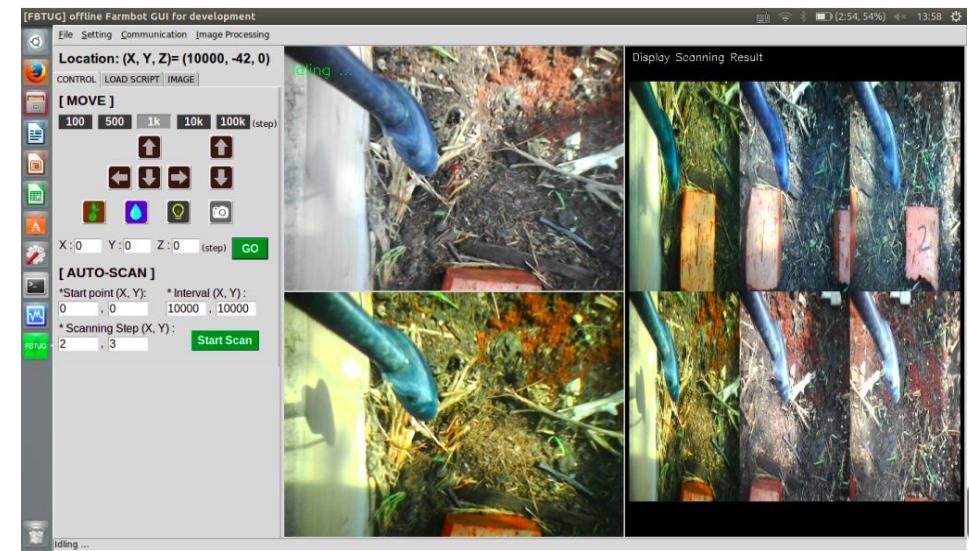
• Ver 1.0



• Ver 2.0



• Ver 3.0

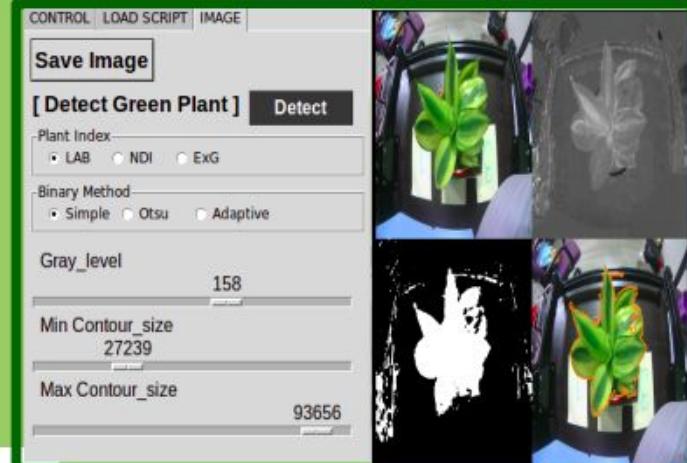


FBTUG Commander

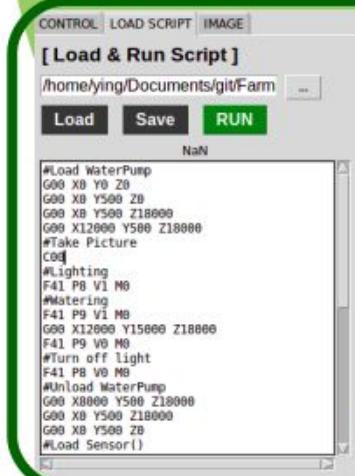
General Control



Detect Green Plant



Commander Function



Load UTM



Auto Watering

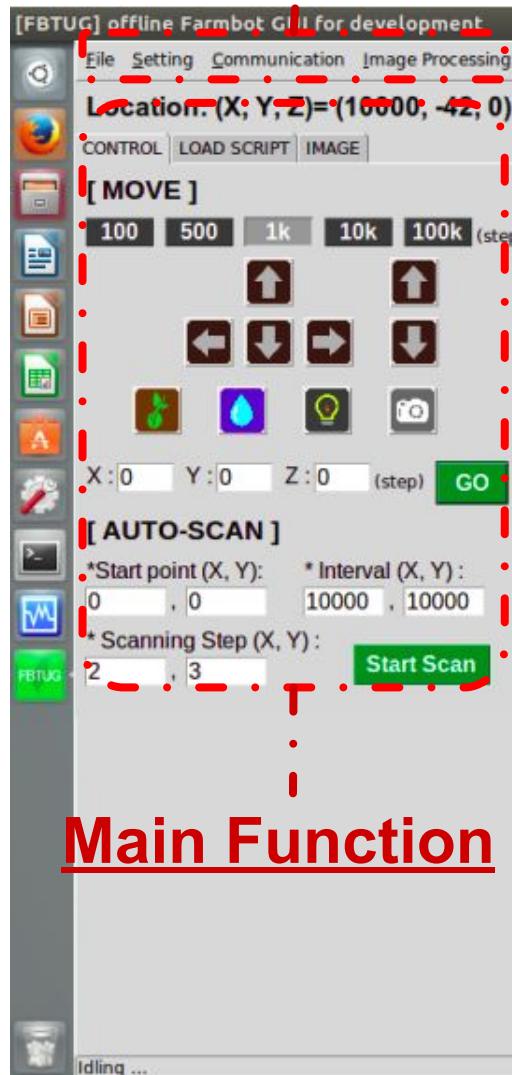


Watering & Lighting

3TUG

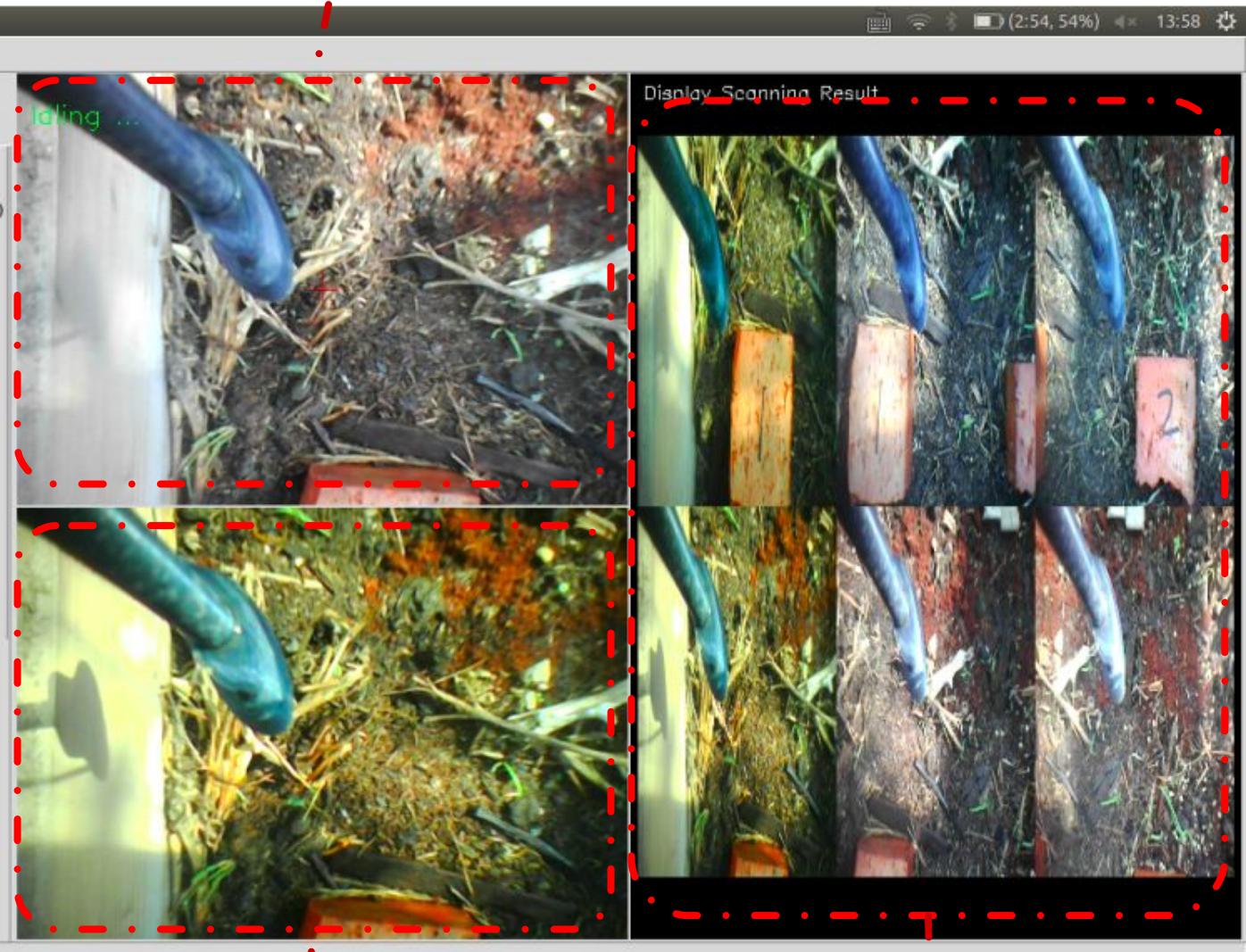
FBTUG Commander

Menu Bar



Main Function

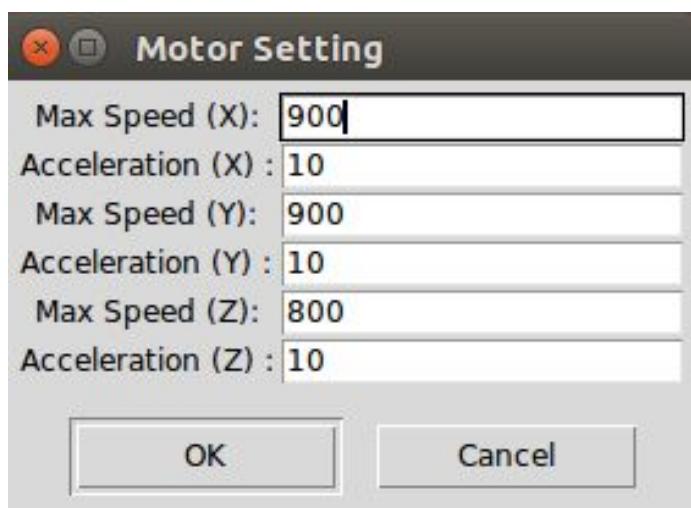
Real-time Image



Process Result

Scanning Result

FBTUG Commander – Menu Bar



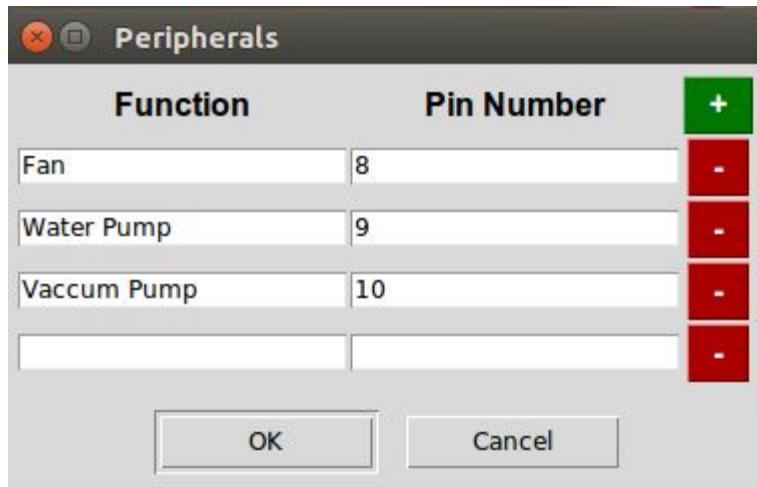
Camera Setting

- Camera ID: usually the ID is 0 with only one camera

Motor Setting

- Max Speed (X)
- Acceleration (X)
- Max Speed (Y)
- Acceleration (Y)
- Max Speed (Z)
- Acceleration (Z)

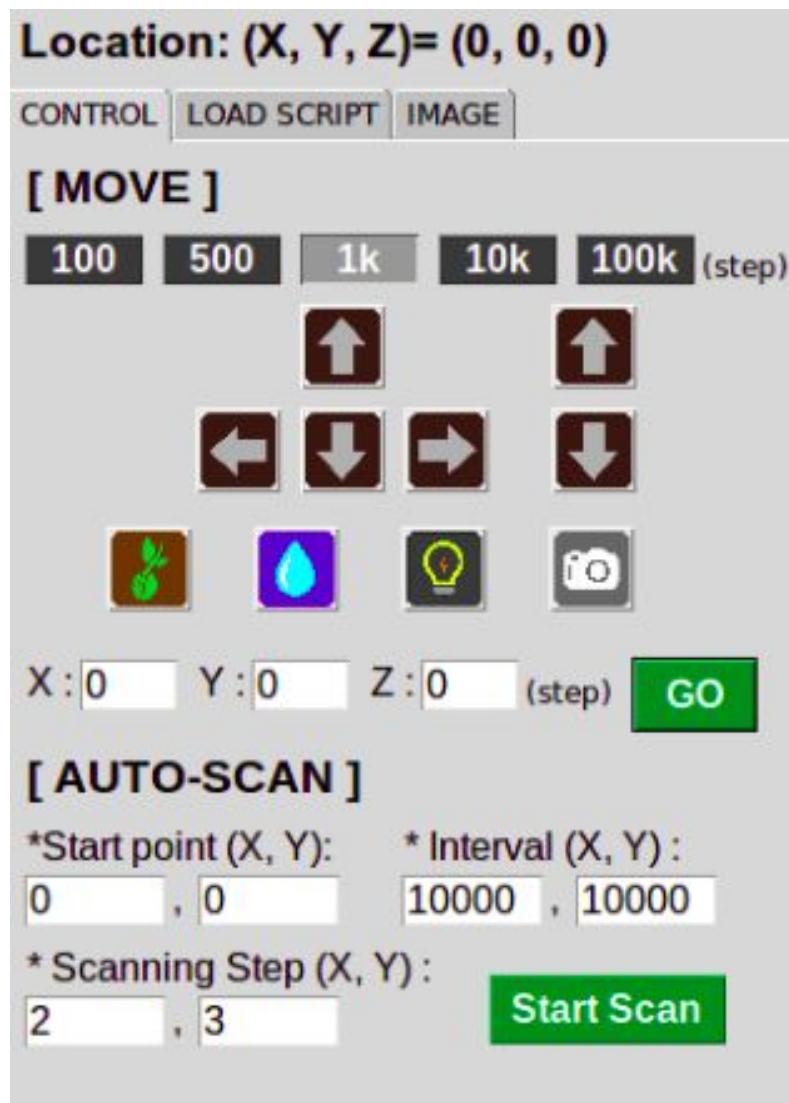
FBTUG Commander – Menu Bar



Peripheral Setting

- Fan → for cooling RPi
- Water Pump → for watering
- Vacuum Pump → for seeding

FBTUG Commander – General Control



Basic function to control farmbot

- Relative movement
 - *Moving Interval*: 100, 500, 1k, 10, 100k (step)
Hot key: **F1 ~ F5**
 - *Moving Direction*: +/-X, +/-Y, +/-Z
Hot key: $\uparrow, \downarrow, \leftarrow, \rightarrow, \text{ctrl}+\uparrow, \text{ctrl}+\downarrow$
- Seeding, Watering, Lighting, Grabbing image
- Absolute movement
 - *X, Y, Z position*
- Auto-Sacnning Setting
 - *Start Point (X, Y)*
 - *Scanning interval (X, Y)*
 - *Scanning Step (X, Y)*



FBTUG Commander – G,F code

Location: (X, Y, Z)= (0, 0, 0)

CONTROL LOAD SCRIPT IMAGE

[Load & Run Script]

/home/ying/Documents/git/Stepp ...

Load Save RUN

- serial_script.txt -

```
#Current Debug
G00 X500 Y0 Z0#Move to location at given
F41 P10 V1 M0
G00 X1000 Y0 Z0#Move to location at give
F41 P10 V0 M0
G00 X1500 Y0 Z0#Move to location at give
G00 X0 Y0 Z0
G00 X1500 Y2000 Z0#Move to location at g
F83#Report software version
```

G code & F code

Code type	Number	Parameters	Function
G			G-Code, the codes working the same as a 3D printer
G	00	X Y Z S	Move to location at given speed for axis (don't have to be a straight line), in absolute coordinates
G	01	X Y Z S	Move to location on a straight line
G	28		Move home all axis
F			Farm commands, commands specially added for the farmbot
F	01	T	Dose amount of water using time in millisecond
F	02	N	Dose amount of water using flow meter that measures pulses
F	09		Reset emergency stop
F	11		Home X axis
F	12		Home Y axis
F	13		Home Z axis
F	14		Calibrate X axis
F	15		Calibrate Y axis
F	16		Calibrate Z axis
F	20		List all parameters and value

Reference:

<https://github.com/FarmBot/farmbot-arduino-firmware>



FBTUG Commander – G,F code

Location: (X, Y, Z)= (0, 0, 0)

CONTROL LOAD SCRIPT IMAGE

[Load & Run Script]

/home/ying/Documents/git/Stepp ...

Load Save RUN

- serial_script.txt -

```
#Current Debug
G00 X500 Y0 Z0#Move to location at given
F41 P10 V1 M0
G00 X1000 Y0 Z0#Move to location at give
F41 P10 V0 M0
G00 X1500 Y0 Z0#Move to location at give
G00 X0 Y0 Z0
G00 X1500 Y2000 Z0#Move to location at g
F83#Report software version
```

G code & F code example

e.g. Moving 100, 200, 300 steps on X & Y
& Z direction respectively:

→ G00 X100 Y200 Z300

e.g. Read the Parameter Value of X
acceleration

→ F21 P41

e.g. Change the Parameter Value of Y
acceleration to 300

→ F22 P41 V300

•

•



FBTUG Commander – C code Script

Location: (X, Y, Z)= (0, 0, 0)

CONTROL LOAD SCRIPT IMAGE

[Load & Run Script]

/home/ying/Documents/git/Stepp ...

Load Save RUN

- serial_script.txt -

```
#Current Debug
G00 X500 Y0 Z0#Move to location at given
F41 P10 V1 M0
G00 X1000 Y0 Z0#Move to location at give
F41 P10 V0 M0
G00 X1500 Y0 Z0#Move to location at give
G00 X0 Y0 Z0
G00 X1500 Y2000 Z0#Move to location at g
F83#Report software version
```

Code sent to RPI (defined by FBTUG)

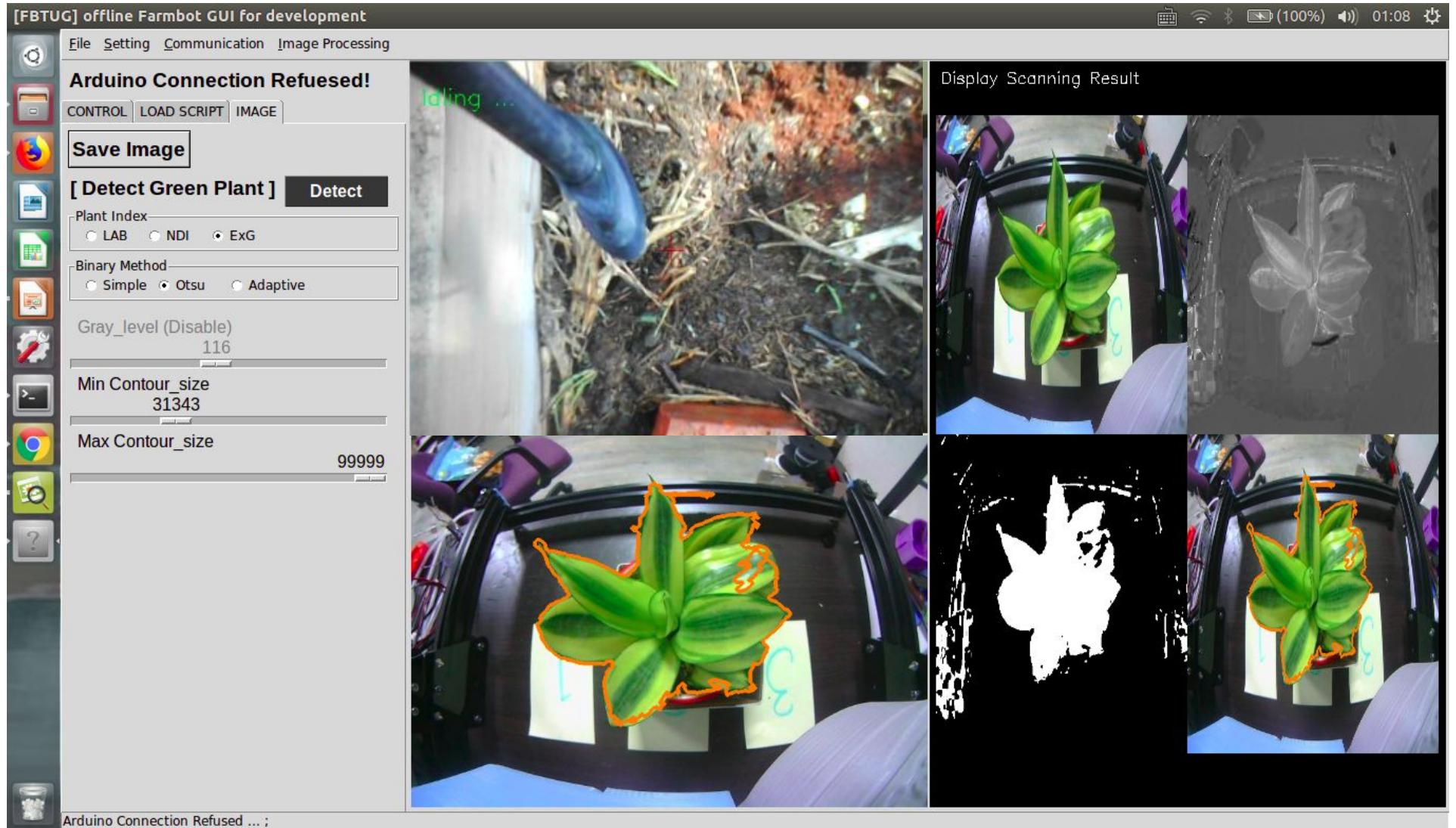
Code type	Number	Parameters	Function
C			Camera command, commands for controlling camera
C	00		Save image
T	00	h m s	(Not added yet) Delay h hours, m mintus, s seconds
T	01	h m s	(Not added yet) Wait until it is at hM ^s

→ C00

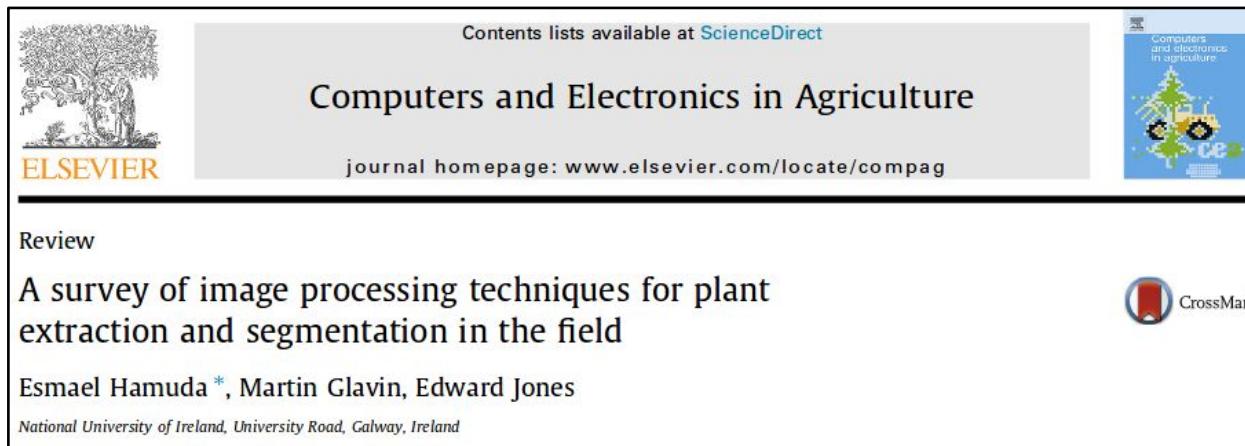
(Directory: Data/YmdHMS_X_Y_X.png)



FBTUG Commander – Green Plant Detection



FBTUG Commander – Green Plant Detection



- **Normalized Difference Index (NDI)**

$$NDI = 128 * \left(\left(\frac{(G - R)}{(G + R)} \right) + 1 \right)$$

- **Excess Green Index (ExG)**

$$\text{ExG} = 2g - r - b$$
$$r = \frac{R^*}{(R^* + G^* + B^*)}, \quad g = \frac{G^*}{(R^* + G^* + B^*)}, \quad b = \frac{B^*}{(R^* + G^* + B^*)}$$
$$R^* = \frac{R}{R_{max}}, \quad G^* = \frac{G}{G_{max}}, \quad B^* = \frac{B}{B_{max}} \quad R_{max} = G_{max} = B_{max} = 255$$

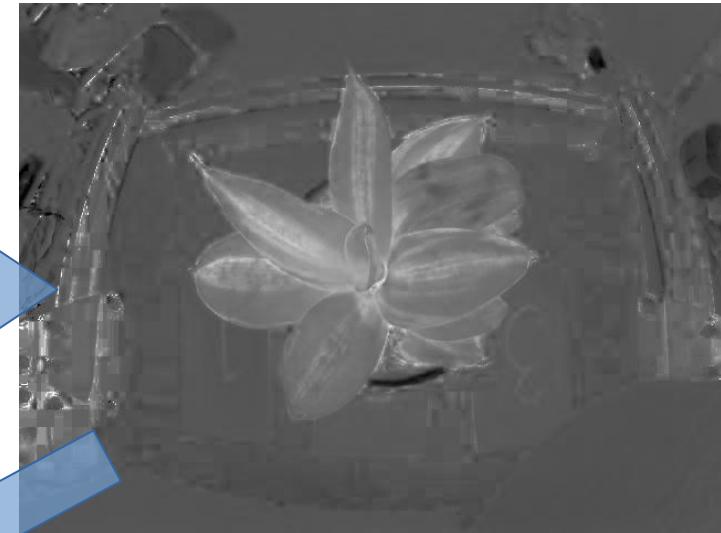
Reference: <http://agri.ckcest.cn/ass/NK006-20160808005.pdf>

FBTUG Commander – Green Plant Detection

. Raw Image



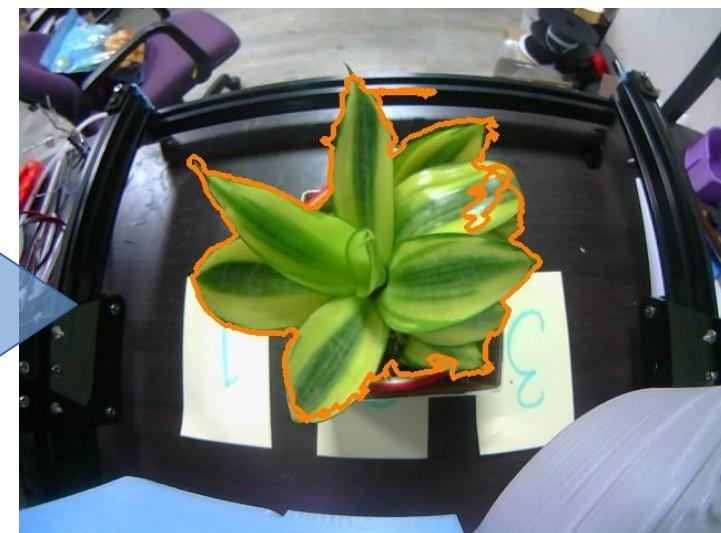
. Plant Index



. Binary Image



. Contour Extraction



FBTUG

Summary

Farmbot Web App	FBTUG Commander
<ul style="list-style-type: none">• Commercial Product• Lower barrier for End User• Easy to maintenance• Remote Control <ul style="list-style-type: none">• Load and run script of G&F&C code to complete sequence actions• Auto scanning function• Real-time image display• Detect Green Plant function(LAB, ExG and NDI method are provided here)	<ul style="list-style-type: none">• Development Tool• Lower barrier for Maker• Easy to debug• Flexible



Next Step

- T code for Time schedule function
- Calibration Function
- Update data to server
- Image Process Tool
- Log event function
- AI testing platform
-



Example of UI in Python



Practice

https://github.com/SpongeYao/Maker_Practice

SpongeYao / Maker_Practice

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

An example of Tkinter UI and simply finding green object by HSV color space Edit

Manage topics

3 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download ▾

SpongeYao ver0 README.md Initial commit 3 days ago

openCV_FindGreen.py ver0 3 days ago

openCV_FindGreen_practice.py ver0 3 days ago

test_findGreen.jpg ver0 3 days ago

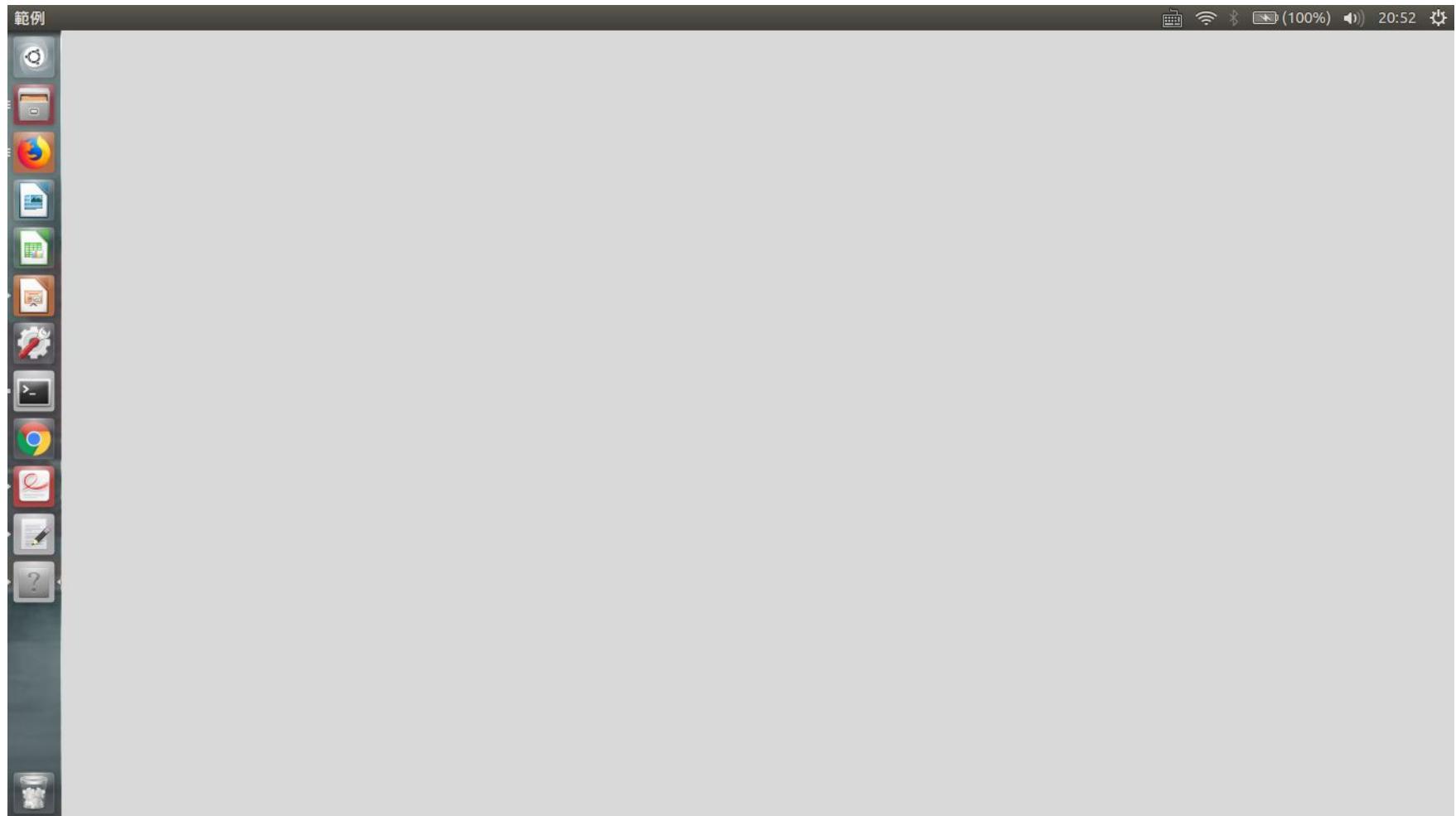
README.md

Maker_Practice

The image shows a GitHub repository page for 'Maker_Practice'. At the top, there's a navigation bar with links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. Below the navigation bar, a main heading says 'An example of Tkinter UI and simply finding green object by HSV color space' with an 'Edit' button. There's also a 'Manage topics' link. Underneath, there are summary stats: '3 commits', '1 branch', '0 releases', and '1 contributor'. On the right side of the stats, there's a green 'Clone or download ▾' button, which is highlighted with a red rectangle. Below the stats, there's a dropdown for 'Branch: master' and a 'New pull request' button. Further down, there's a table showing file details: 'README.md' (Initial commit, 3 days ago), 'openCV_FindGreen.py' (ver0, 3 days ago), 'openCV_FindGreen_practice.py' (ver0, 3 days ago), and 'test_findGreen.jpg' (ver0, 3 days ago). At the bottom of the page, there's a section for 'README.md' with an edit icon. The word 'Maker_Practice' is prominently displayed at the bottom left.

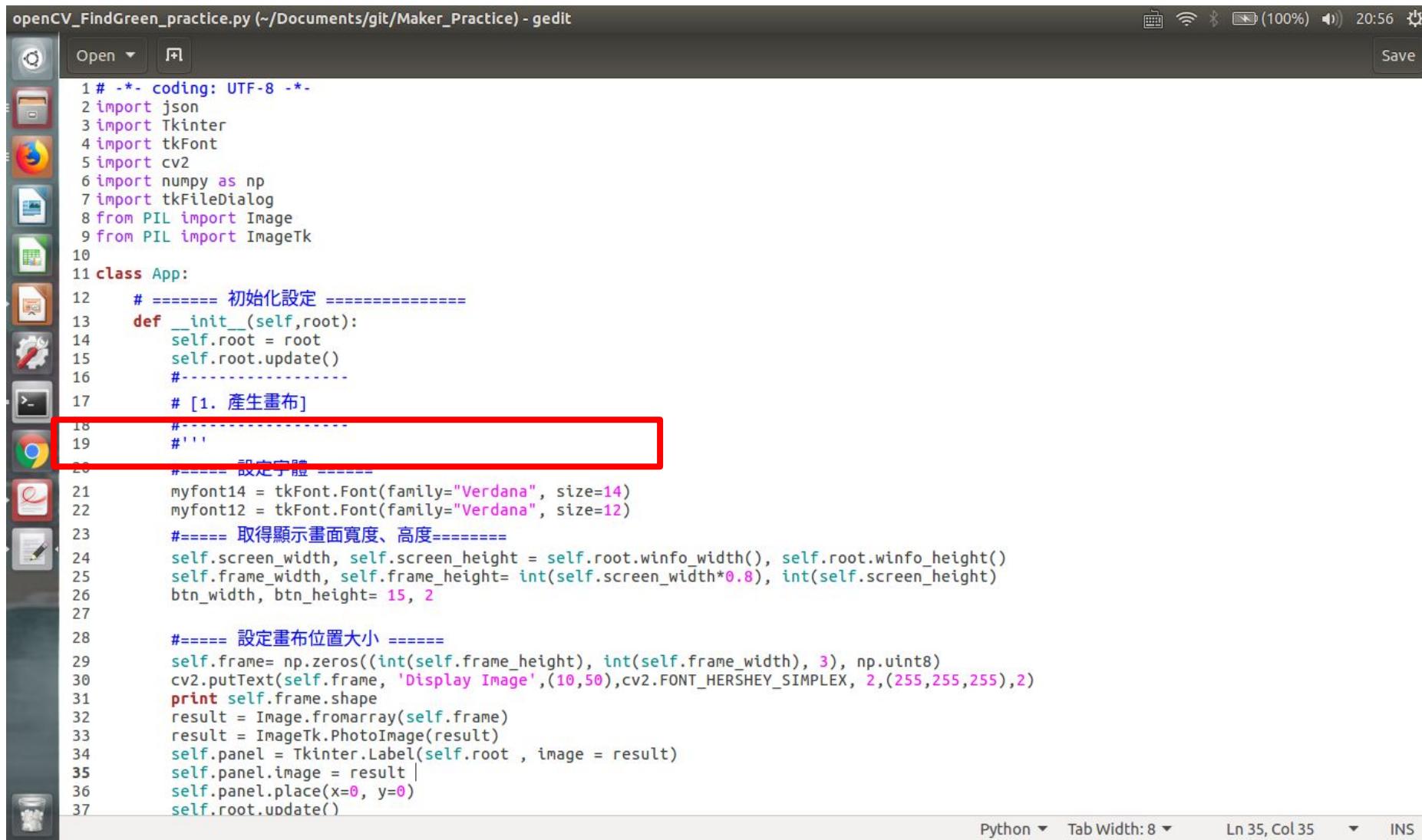
Practice-1

python opencv_FindGreen_practice.py



Practice-2

Add symbol, #, at line 19 before “”



```
openCV_FindGreen_practice.py (~/Documents/git/Maker_Practice) - gedit
Open Save
1 # -*- coding: UTF-8 -*-
2 import json
3 import Tkinter
4 import tkFont
5 import cv2
6 import numpy as np
7 import tkFileDialog
8 from PIL import Image
9 from PIL import ImageTk
10
11 class App:
12     # ===== 初始化設定 =====
13     def __init__(self,root):
14         self.root = root
15         self.root.update()
16         #-----
17         # [1. 產生畫布]
18         #-----
19         #''
20         #---- 設定字體 ----
21         myfont14 = tkFont.Font(family="Verdana", size=14)
22         myfont12 = tkFont.Font(family="Verdana", size=12)
23         #==== 取得顯示畫面寬度、高度=====
24         self.screen_width, self.screen_height = self.root.winfo_width(), self.root.winfo_height()
25         self.frame_width, self.frame_height= int(self.screen_width*0.8), int(self.screen_height)
26         btn_width, btn_height= 15, 2
27
28         #==== 設定畫布位置大小 ====
29         self.frame= np.zeros((int(self.frame_height), int(self.frame_width), 3), np.uint8)
30         cv2.putText(self.frame, 'Display Image',(10,50),cv2.FONT_HERSHEY_SIMPLEX, 2,(255,255,255),2)
31         print self.frame.shape
32         result = Image.fromarray(self.frame)
33         result = ImageTk.PhotoImage(result)
34         self.panel = Tkinter.Label(self.root , image = result)
35         self.panel.image = result |
36         self.panel.place(x=0, y=0)
37         self.root.update()
```

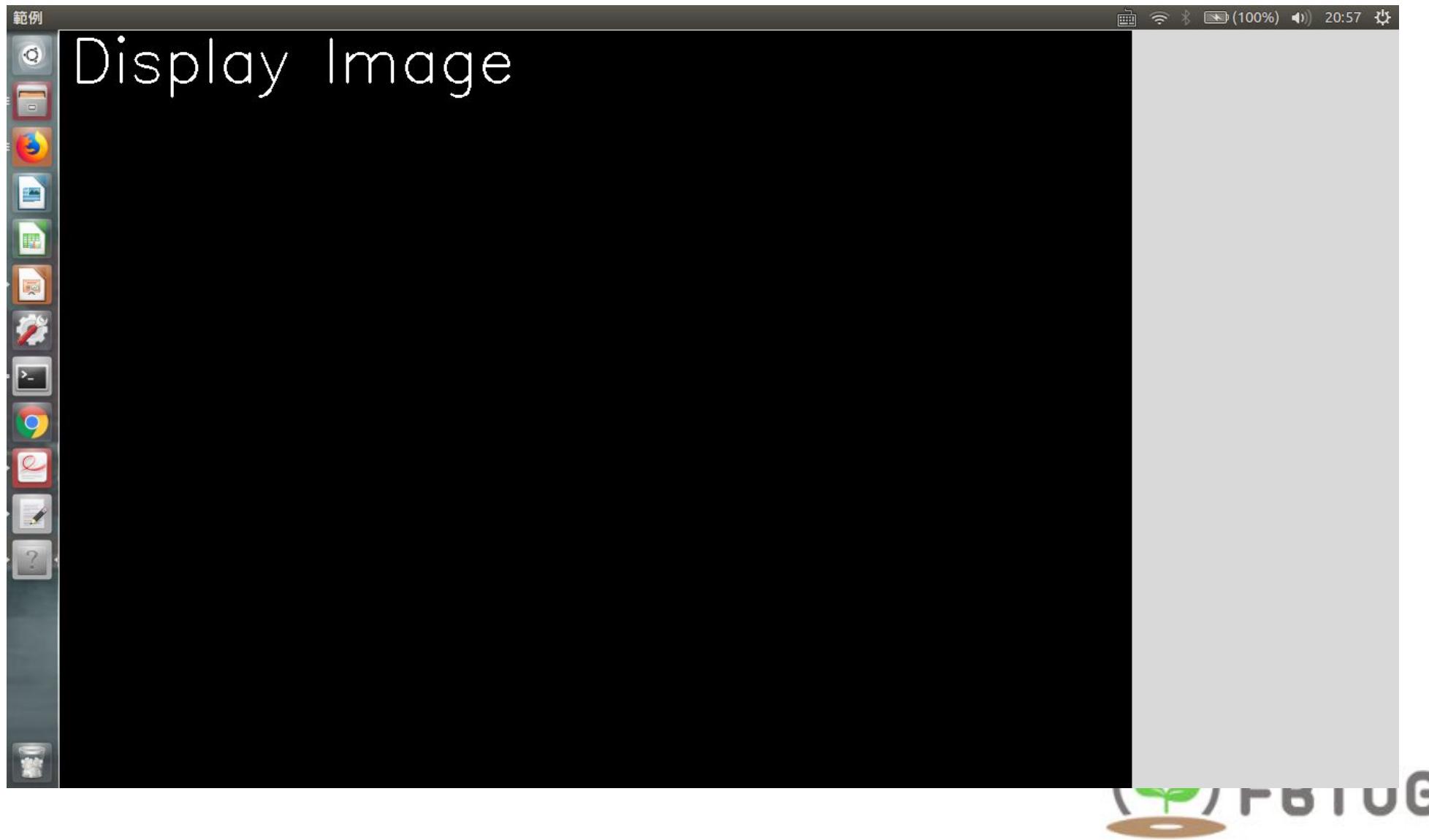
Python ▾ Tab Width: 8 ▾

Ln 35, Col 35

INS

Practice-2

python opencv_FindGreen_practice.py



Practice-3

Add symbol, #, at line 43 before “”

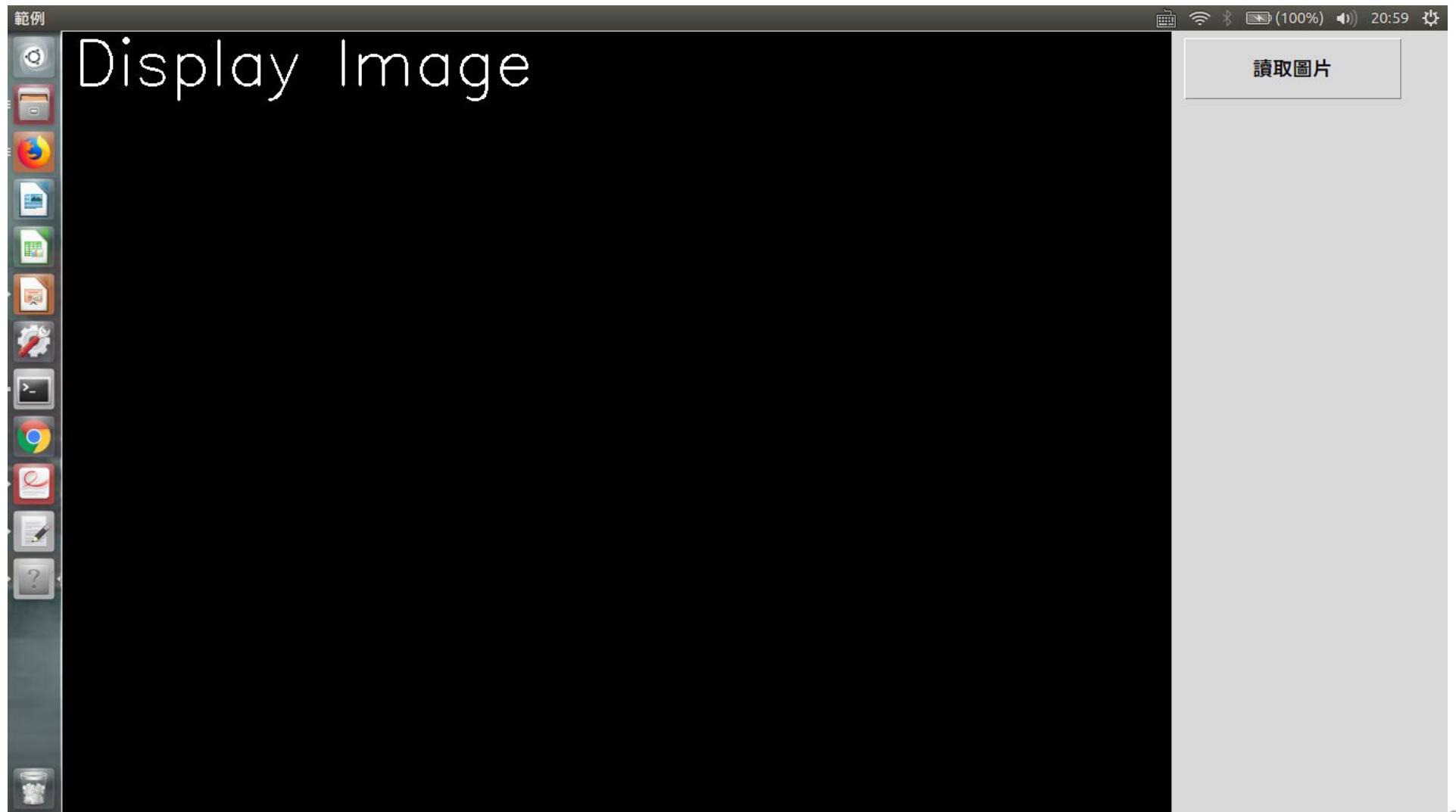
```
*openCV_FindGreen_practice.py (~/Documents/git/Maker_Practice) - gedit
Open Save
30     cv2.putText(self.frame, 'Display Image',(10,50),cv2.FONT_HERSHEY_SIMPLEX, 2,(255,255,255),2)
31     print self.frame.shape
32     result = Image.fromarray(self.frame)
33     result = ImageTk.PhotoImage(result)
34     self.panel = Tkinter.Label(self.root , image = result)
35     self.panel.image = result
36     self.panel.place(x=0, y=0)
37     self.root.update()
38     #'''
39
40     #-----
41     # [2. 產生讀取圖片按鈕]
42     #-----
43     #''' # ===== 設定按鈕位置大小 =====
44     # ===== 設定按鈕位置大小 =====
45     self.btn_loadImg= Tkinter.Button(self.root, text= '讀取圖片', command= self.btn_loadImg_click,font= myfont14, width= btn_width,
height=btn_height)
46     self.btn_loadImg.place(x= self.frame_width+12, y=6)
47     self.root.update()
48     #'''
49     #-----
50     # [3. 產生尋找按鈕]
51     #-----
52     #''' # ===== 設定按鈕位置大小 =====
53     # ===== 設定按鈕位置大小 =====
54     self.btn_saveImg= Tkinter.Button(self.root, text= '儲存圖片', command= self.btn_saveImg_click,font= myfont14, width= btn_width,
height=btn_height)
55     self.btn_saveImg.place(x= self.btn_loadImg.winfo_x(), y=self.btn_loadImg.winfo_y()+self.btn_loadImg.winfo_height()+ 6)
56     self.root.update()
57     #'''
58     #-----
59     # [4. 產生二值化按鈕]
60     #-----
61     #'''
```

Python ▾ Tab Width: 8 ▾ Ln 42, Col 9 ▾ INS



Practice-3

python opencv_FindGreen_practice.py



Practice-4

Add symbol, #, at line 52 before “”

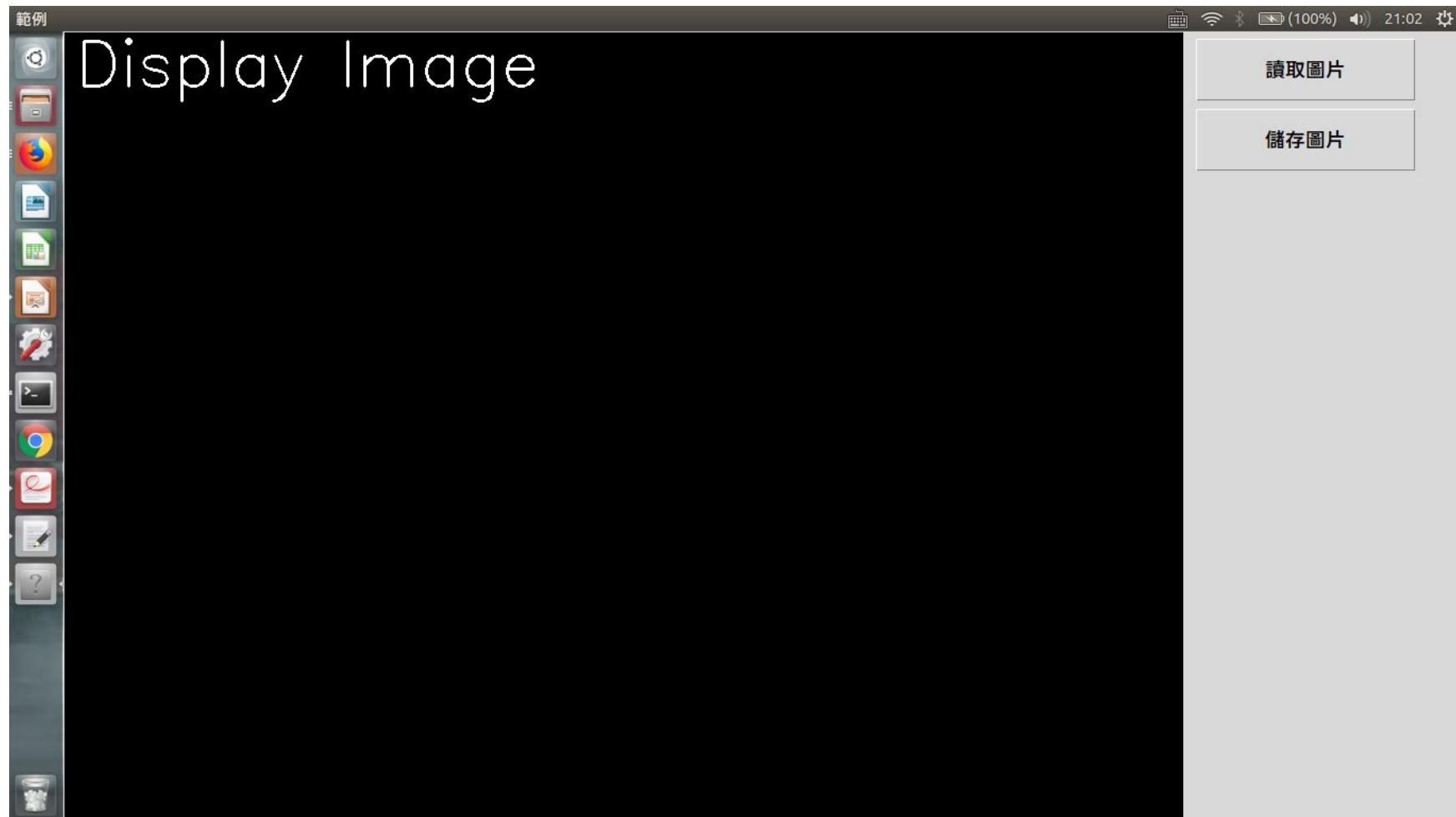
```
*openCV_FindGreen_practice.py (~/Documents/git/Maker_Practice) - gedit
Open Save
39
40      # -----
41      # [2. 產生讀取圖片按鈕]
42      # -----
43      # ''
44      # ===== 設定按鈕位置大小 =====
45      self.btn_loadImg= Tkinter.Button(self.root, text= '讀取圖片', command= self.btn_loadImg_click,font= myfont14, width= btn_width,
height=btn_height)
46      self.btn_loadImg.place(x= self.frame_width+12, y=6)
47      self.root.update()
48      # ''
49      # -----
50      # [3. 產生尋找按鈕]
51      # -----
52      # ''# ===== 改走按鈕位置大小 =====
53      # -----
54      self.btn_saveImg= Tkinter.Button(self.root, text= '儲存圖片', command= self.btn_saveImg_click,font= myfont14, width= btn_width,
height=btn_height)
55      self.btn_saveImg.place(x= self.btn_loadImg.winfo_x(), y=self.btn_loadImg.winfo_y()+self.btn_loadImg.winfo_height()+ 6)
56      self.root.update()
57      # ''
58      # -----
59      # [4. 產生二值化按鈕]
60      # -----
61      # ''
62      # ''
63      # ===== 設定按鈕位置大小 =====
64      self.btn_binarization= Tkinter.Button(self.root, text='二值化', command= self.btn_binarization_click,font= myfont14, width=
btn_width, height= btn_height)
65      self.btn_binarization.place(x= self.btn_saveImg.winfo_x(), y=self.btn_saveImg.winfo_y()+self.btn_saveImg.winfo_height()+ 6)
66      self.root.update()
67      # ''
68      # -----
69      # [5. 產生二值化閾值滾軸]
```

Python ▾ Tab Width: 8 ▾ Ln 52, Col 10 ▾ INS



Practice-4

python opencv_FindGreen_practice.py



Practice-5

Add symbol, #, at line 62 before “”

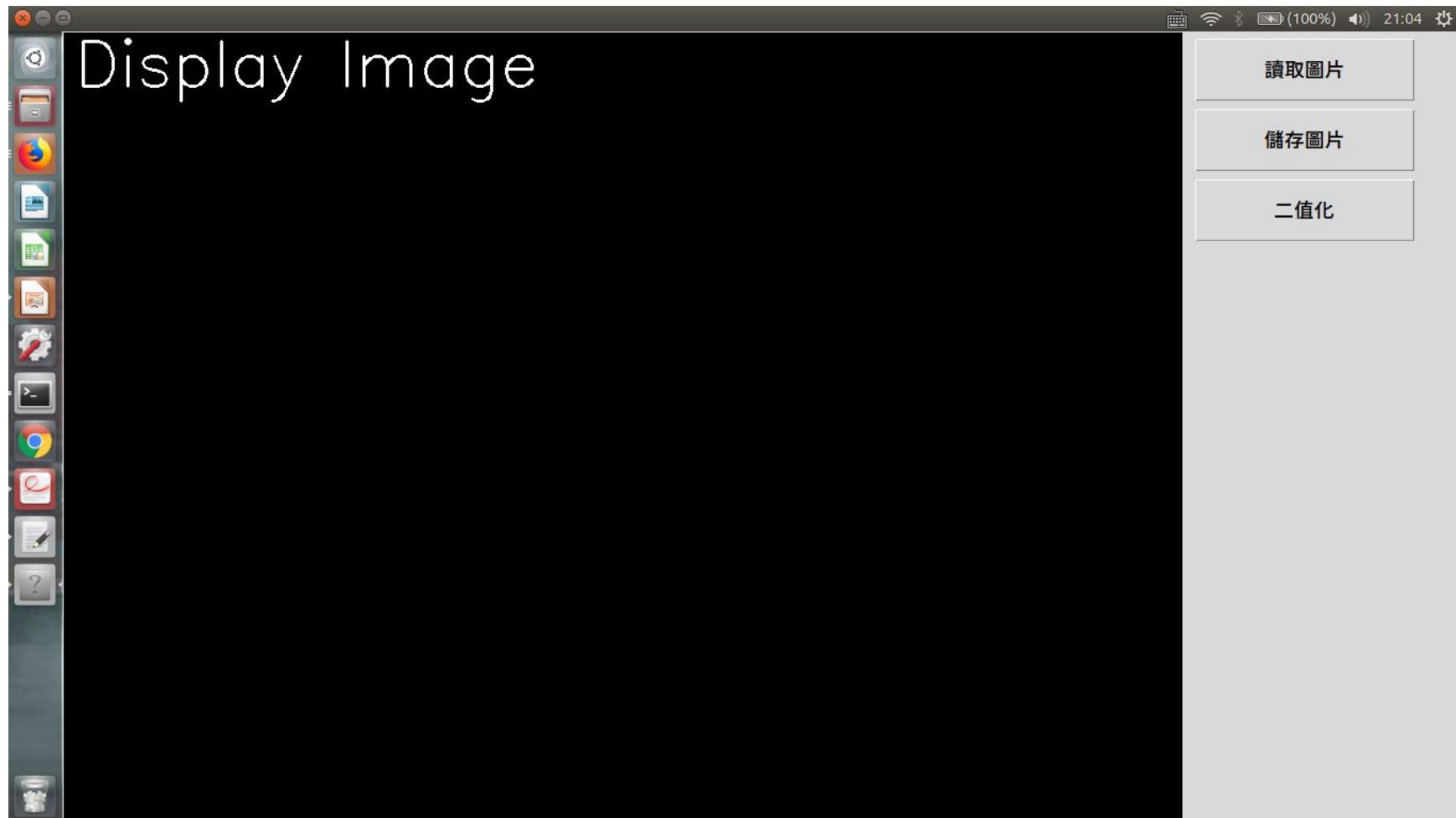
```
*openCV_FindGreen_practice.py (~/Documents/git/Maker_Practice) - gedit
Open Save
50      # [3. 產生尋找按鈕]
51      #-----
52      #''
53      # ===== 設定按鈕位置大小 =====
54      self.btn_saveImg= Tkinter.Button(self.root, text= '儲存圖片', command= self.btn_saveImg_click,font= myfont14, width= btn_width,
height=btn_height)
55      self.btn_saveImg.place(x= self.btn_loadImg.winfo_x(), y=self.btn_loadImg.winfo_y()+self.btn_loadImg.winfo_height()+ 6)
56      self.root.update()
57      #''
58      #-----
59      # [4. 產生二值化按鈕]
60      #-----
61      #''
62      #'' 設定按鈕位置大小
63      # -----
64      self.btn_binarization= Tkinter.Button(self.root, text='二值化', command= self.btn_binarization_click,font= myfont14, width=
btn_width, height= btn_height)
65      self.btn_binarization.place(x= self.btn_saveImg.winfo_x(), y=self.btn_saveImg.winfo_y()+self.btn_saveImg.winfo_height()+ 6)
66      self.root.update()
67      #''
68      #-----
69      # [5. 產生二值化閾值滾軸]
70      #-----
71      ...
72      self.threshold_graylevel= 125
73      self.scale_threshold_graylevel = Tkinter.Scale(self.root , from_= 0 , to = 255 , orient = Tkinter.HORIZONTAL , label = "二值化閾值
(0~255) ", font = myfont12, width = 7, length = 200 )
74      self.scale_threshold_graylevel.set(self.threshold_graylevel)
75      self.scale_threshold_graylevel.place(x= self.btn_binarization.winfo_x(), y=self.btn_binarization.winfo_y()
+self.btn_binarization.winfo_height()+ 6)
76      self.root.update()
77      #''
78      #-----
79      # [6. 產生尋找綠色按鈕]
```

Python ▾ Tab Width: 8 ▾ Ln 62, Col 10 ▾ INS



Practice-5

python opencv_FindGreen_practice.py



Practice-6

Add symbol, #, at line 71 before “”

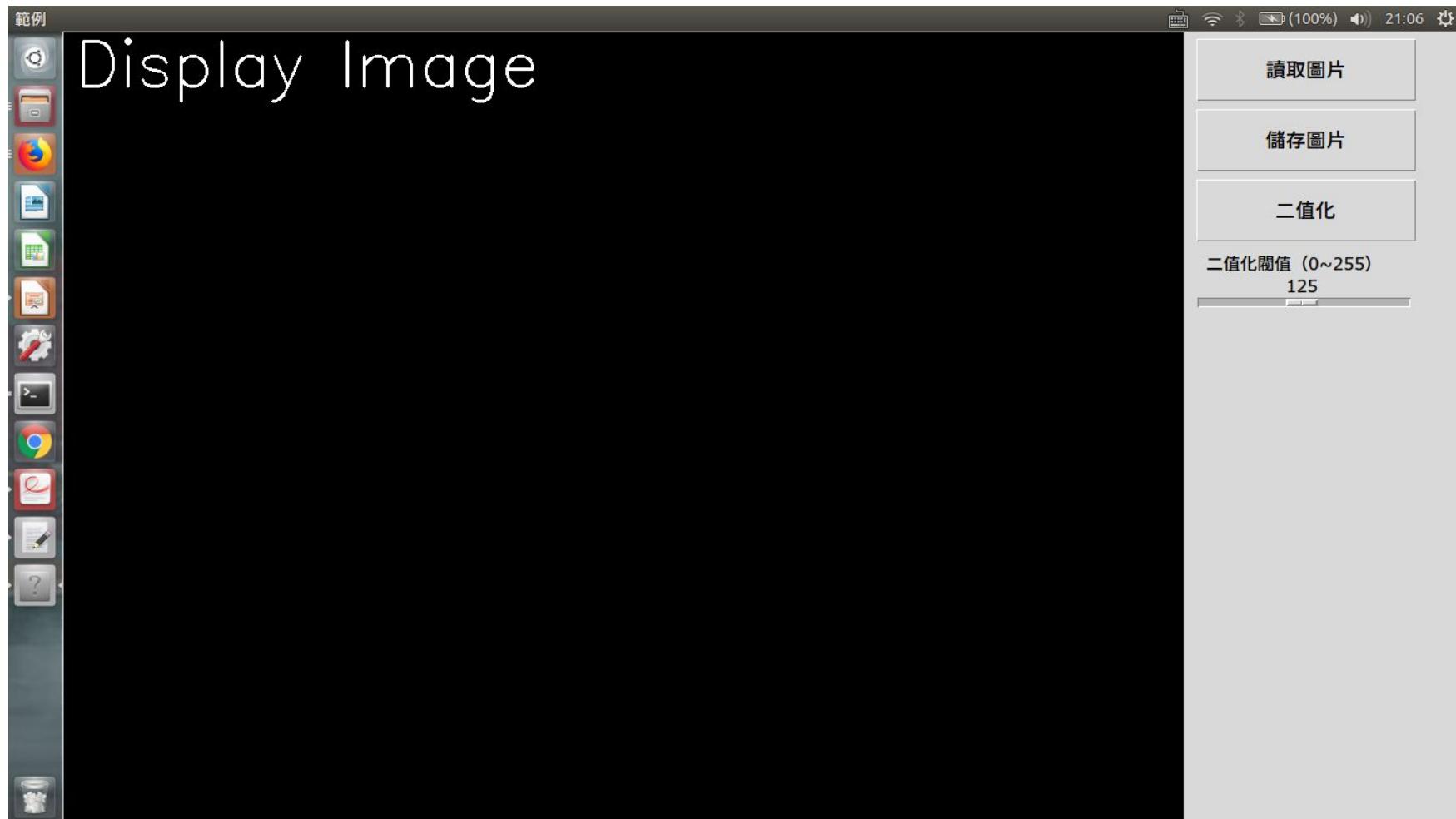
```
*openCV_FindGreen_practice.py (~/Documents/git/Maker_Practice) - gedit
Open + Save
60      # [4. 產生二值化按鈕]
61      #
62      #'''
63      # ===== 設定按鈕位置大小 ======
64      self.btn_binarization= Tkinter.Button(self.root, text='二值化', command= self.btn_binarization_click,font= myfont14, width=
65          btn_width, height= btn_height)
66      self.btn_binarization.place(x= self.btn_saveImg.winfo_x(), y=self.btn_saveImg.winfo_y()+self.btn_saveImg.winfo_height()+ 6)
67      self.root.update()
68      #'''
69      # -----
70      # [5. 產生二值化閾值滾軸]
71      #'''
72      self.threshold_qraylevel= 125
73      self.scale_threshold_graylevel = Tkinter.Scale(self.root , from_= 0 , to = 255 , orient = Tkinter.HORIZONTAL , label = "二值化閾值
74          (0~255) ", font = myfont12, width = 7, length = 200 )
75      self.scale_threshold_graylevel.set(self.threshold_qraylevel)
76      self.scale_threshold_graylevel.place(x= self.btn_binarization.winfo_x(), y=self.btn_binarization.winfo_y()
77          +self.btn_binarization.winfo_height()+ 6)
78      self.root.update()
79      #'''
80      # -----
81      # [6. 產生尋找綠色按鈕]
82      #
83      # ===== 設定按鈕位置大小 ======
84      self.btn_findGreen= Tkinter.Button(self.root, text='尋找綠色', command= self.btn_findGreen_click,font= myfont14, width= btn_width,
85          height= btn_height)
86      self.btn_findGreen.place(x= self.scale_threshold_graylevel.winfo_x(), y=self.scale_threshold_graylevel.winfo_y()
87          +self.scale_threshold_graylevel.winfo_height()+ 6)
88      self.root.update()
89      #'''
90      # ===== 顯示圖片至畫布 ======
```

Python ▾ Tab Width: 8 ▾ Ln 71, Col 10 ▾ INS



Practice-6

python opencv_FindGreen_practice.py



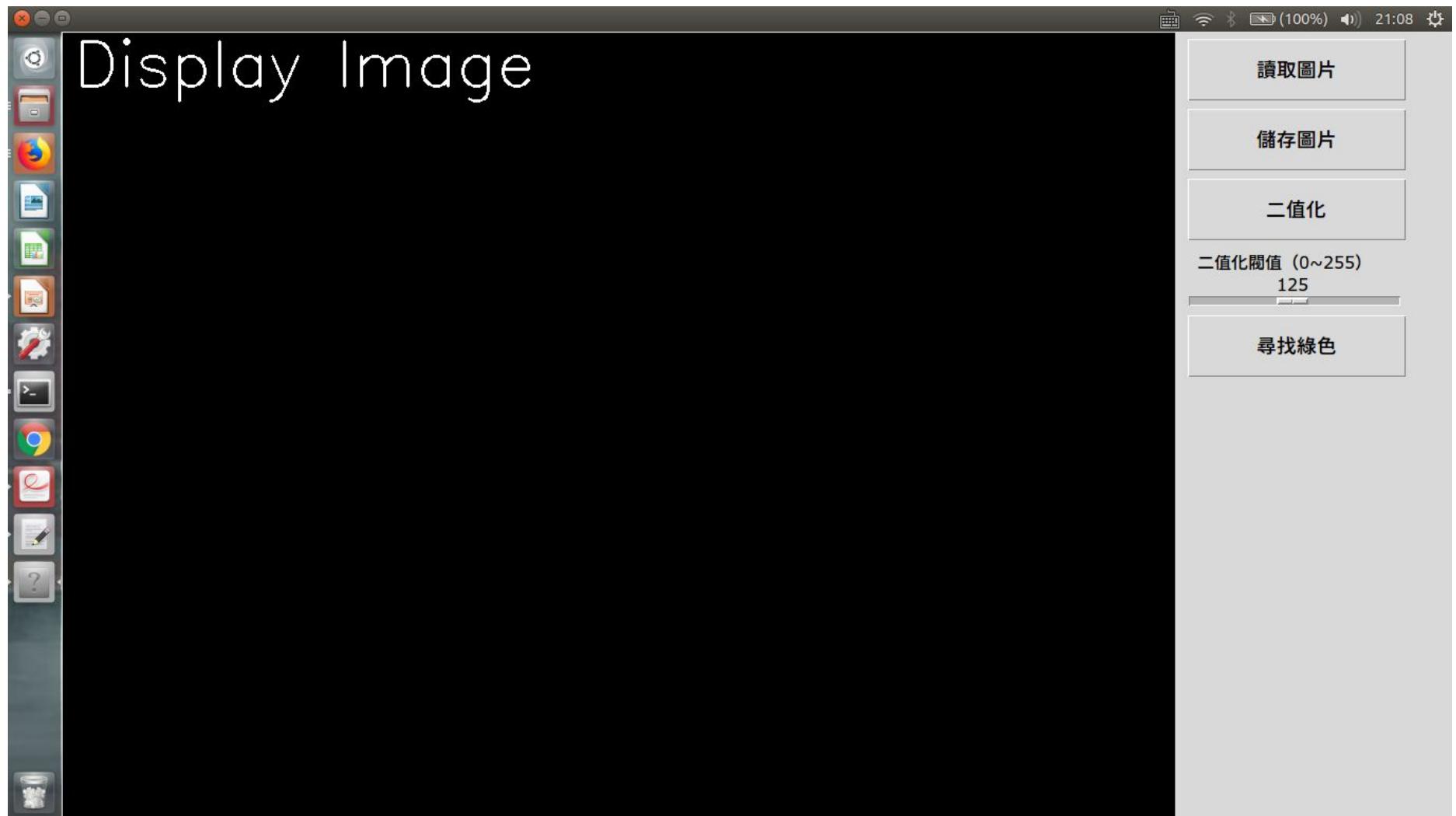
Practice-7

Add symbol, #, at line 81 before “”

```
*openCV_FindGreen_practice.py (~/Documents/git/Maker_Practice) - gedit
Open Save
67      #'''
68      #-----
69      # [5. 產生二值化閾值滾軸]
70      #-----
71      #'''
72      self.threshold_graylevel= 125
73      self.scale_threshold_graylevel = Tkinter.Scale(self.root , from_= 0 , to = 255 , orient = Tkinter.HORIZONTAL , label = "二值化閾值
(0~255)" , font = myfont12, width = 7, length = 200 )
74      self.scale_threshold_graylevel.set(self.threshold_graylevel)
75      self.scale_threshold_graylevel.place(x= self.btn_binarization.winfo_x(), y=self.btn_binarization.winfo_y()
+self.btn_binarization.winfo_height()+ 6)
76      self.root.update()
77      #'''
78      #-----
79      # [6. 產生尋找綠色按鈕]
80      #-----
81      #'''
82      # ===== 改正按鈕位直大小 =====
83      self.btn_findGreen= Tkinter.Button(self.root, text='尋找綠色', command= self.btn_findGreen_click,font= myfont14, width= btn_width,
height= btn_height)
84      self.btn_findGreen.place(x= self.scale_threshold_graylevel.winfo_x(), y=self.scale_threshold_graylevel.winfo_y()
+scale_threshold_graylevel.winfo_height()+ 6)
85      self.root.update()
86      #'''
87
88      # ===== 顯示圖片至畫布 =====
89      def display_panel(self, arg_frame):
90          if len(arg_frame.shape)==3:
91              tmp_frame= cv2.cvtColor(arg_frame, cv2.COLOR_BGR2RGB)
92          else:
93              tmp_frame= cv2.cvtColor(arg_frame, cv2.COLOR_GRAY2RGB)
94
95          tmp_frame= cv2.resize(tmp_frame,(self.frame_width,self.frame_height),interpolation=cv2.INTER_LINEAR)      #2018.02.20-???
96          result = Image.fromarray(tmp_frame)
97          result = ImageTk.PhotoImage(result)
98
Python ▾ Tab Width: 8 ▾ Ln 81, Col 10 ▾ INS
```

Practice-7

python opencv_FindGreen_practice.py



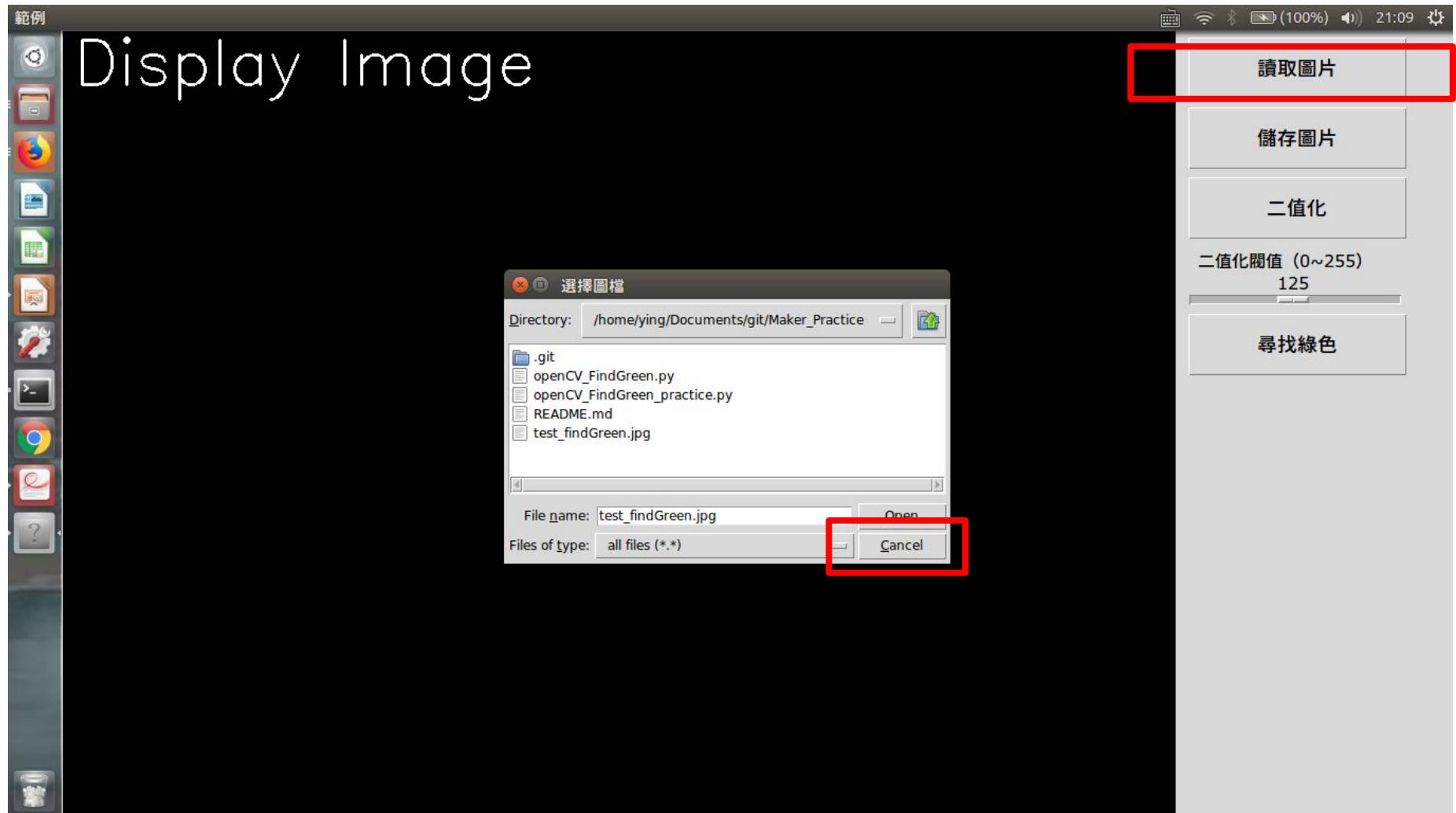
Practice-8

Add the following content after **def btn_loadImg_click(self):**:

```
104     # ===== 讀取圖片 =====
105     def btn_loadImg_click(self):
106         strImagePath = tkFileDialog.askopenfilename(title = "選擇圖
107         檔",filetypes = (("all files","*.*"),("jpeg files","*.jpg"),("png
108         files","*.png")))
109         self.frame= cv2.imread(strImagePath)
110         self.display_panel(self.frame)
111         pass
```

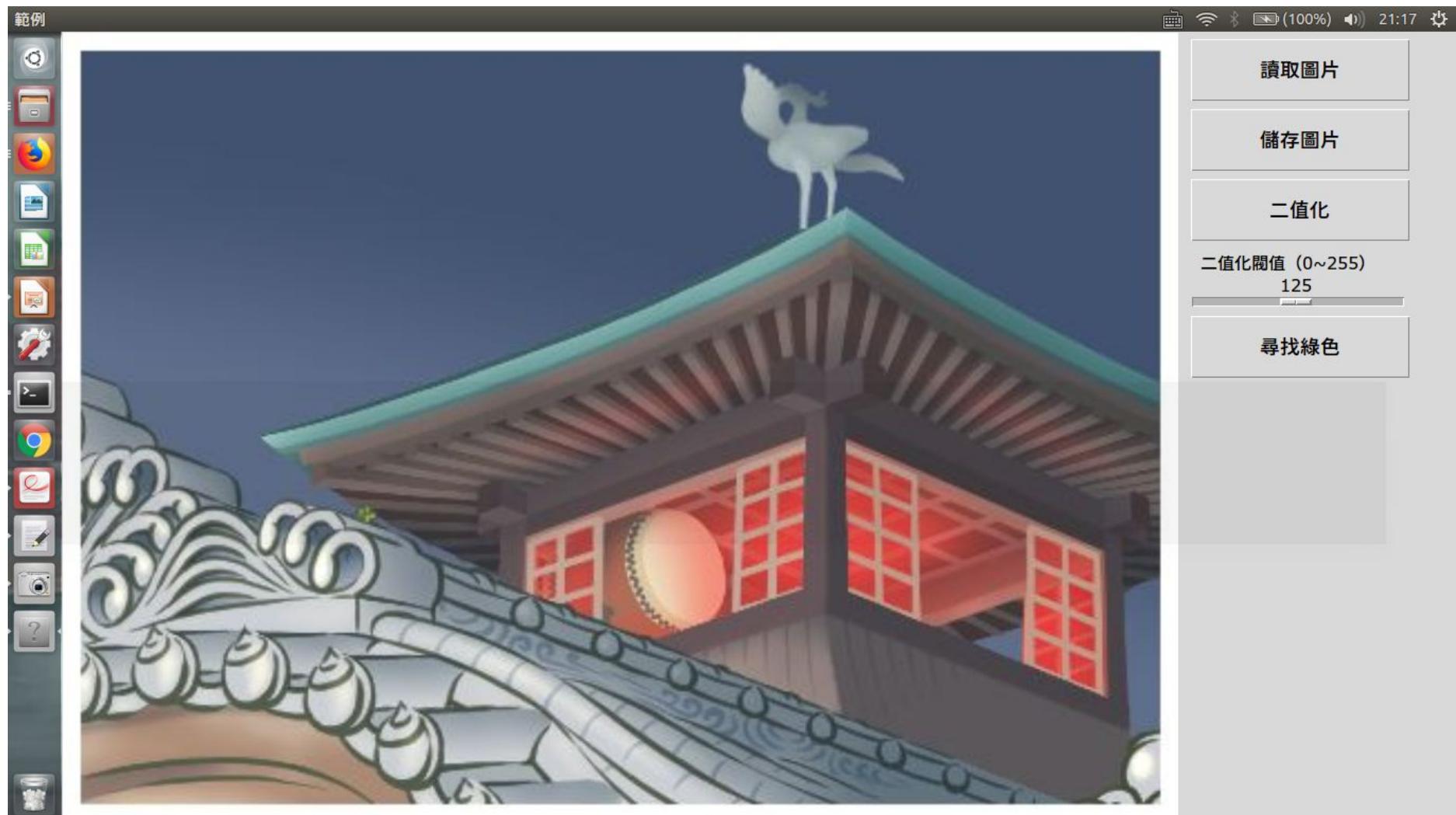
Practice-8

python opencv_FindGreen_practice.py



Practice-8

python opencv_FindGreen_practice.py



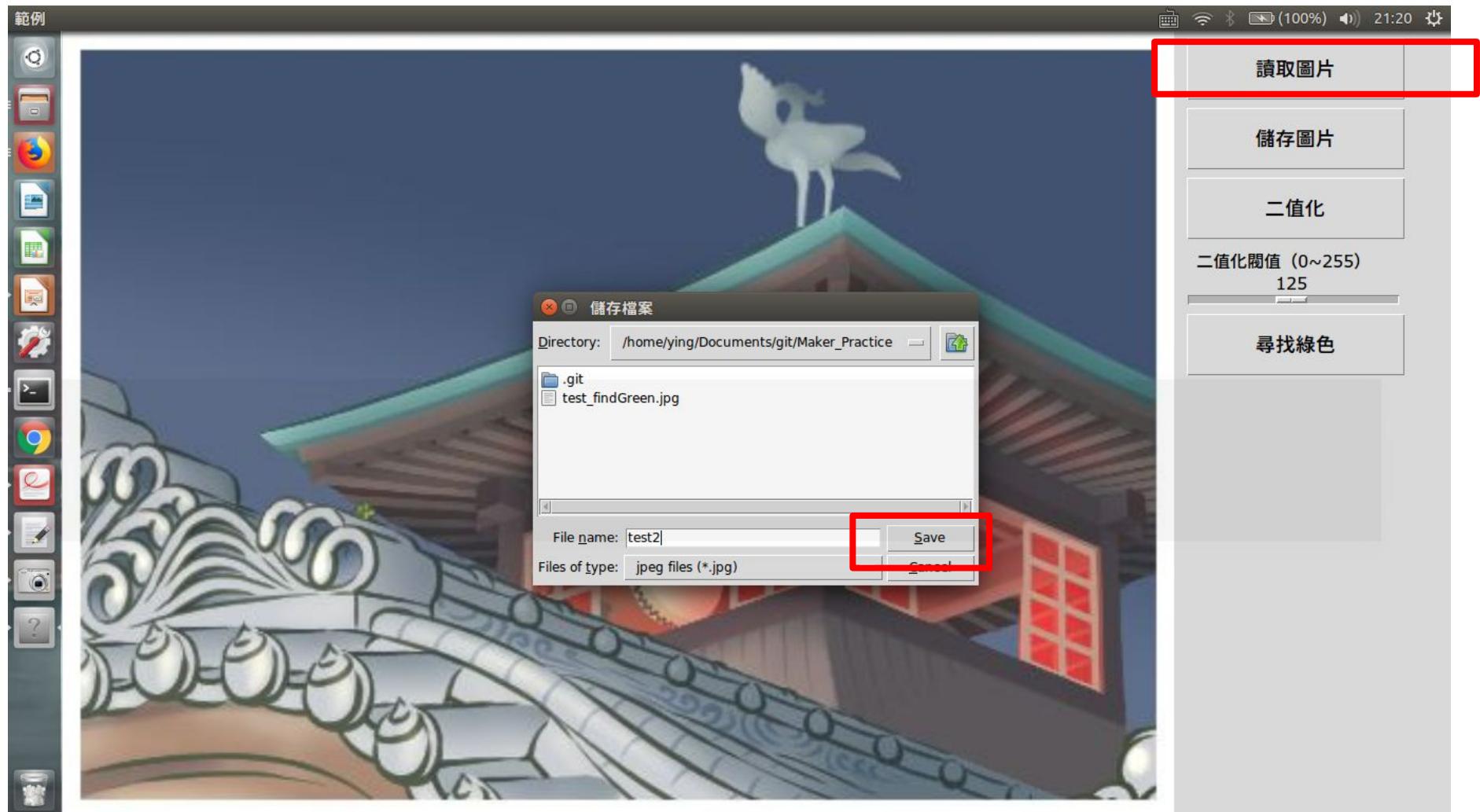
Practice-9

Add the following content after **def btn_saveImg_click(self):**:

```
111     # ===== 儲存圖片 =====
112     def btn_saveImg_click(self):
113         strImagePath = tkFileDialog.asksaveasfilename(title = "儲存
檔案",filetypes = (("jpeg files","*.jpg"),("png files","*.png"),("tif
files","*.tif")))
114         cv2.imwrite(strImagePath,self.frame)
115         pass
116
```

Practice-9

python opencv_FindGreen_practice.py



Practice-10

Add the following content after **def btn_binarization_click(self):**

```
117     # ===== 二值化處理 =====
118     def btn_binarization_click( self ):
119         tmp_frame= cv2.cvtColor(self.frame.copy(),
120             cv2.COLOR_BGR2GRAY)
120         self.threshold_graylevel= self.scale_threshold_graylevel.get()
121         ret, img_thr= cv2.threshold(tmp_frame,
122             self.threshold_graylevel, 255 , 0)
122         self.display_panel( img_thr)
```



Practice-10

數位影像

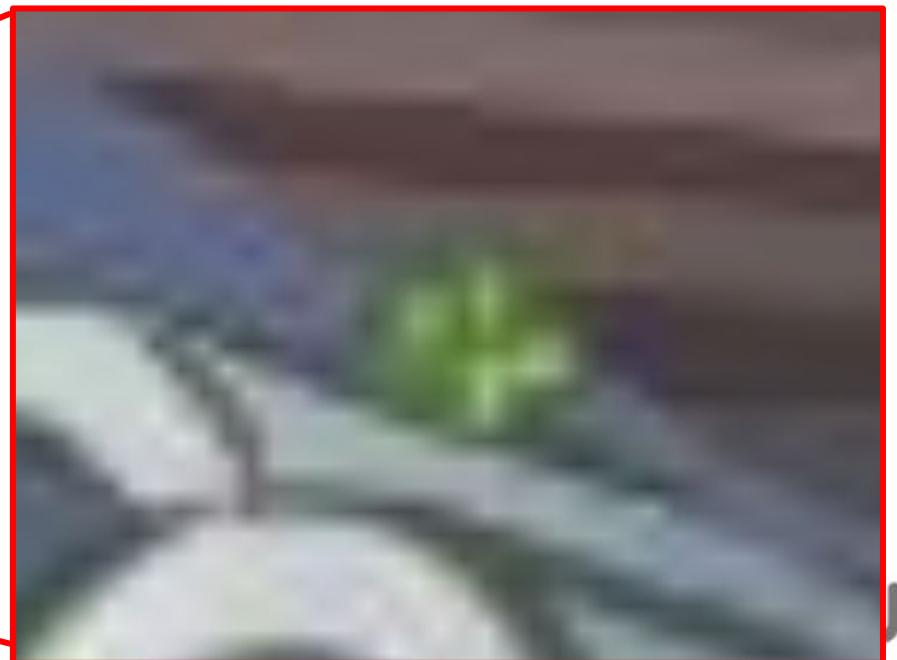
將現實影像做量化，利用有限數值和數量的像素(pixels)來描述二維影像。

Pixel (像素)

影像上的小方格單位

Ex:

200x300 影像 ⇒ 寬 x 高：(200x300)像素



Practice-10

灰階影像(gray image)

單通道(1 channel) 灰階值(gray values)

1 pixel為 1 byte大小。每1 pixel 的值介於 0~255 (256 種顏色)。將亮度量化成 0~255之間：越接近 0代表亮度越低，視覺上呈現黑色；越接近255代表亮度越高，視覺上呈現白色。



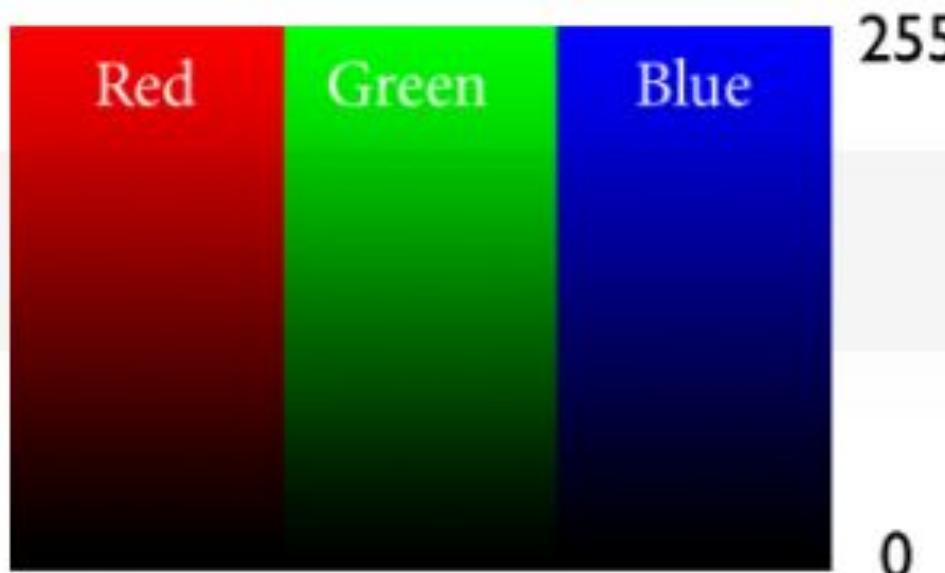
255	255	255	255	255	255	255	255
77	77	77	77	77	77	77	77
255	255	128	128	0	255	255	255
255	255	128	128	0	0	0	255
255	255	128	128	0	0	0	255
255	255	128	128	0	0	0	255
255	255	255	255	255	255	255	255

Practice-10

彩色影像 (color image)

三通道(3 channels) (R, G, B)

將色光三原色做量化，1 pixel為3 bytes大小。即每1 pixel可存的值為 $0 \sim 256^3 - 1 \Rightarrow 0 \sim 0x\text{FFFFFF}$ (pixel values)
意即將三原色各別量化成 $0 \sim 255$ 之間。當一顏色值越接近0，代表該顏色強度越低；越接近255，代表該顏色強度越高。



Practice-10

python opencv_FindGreen_practice.py



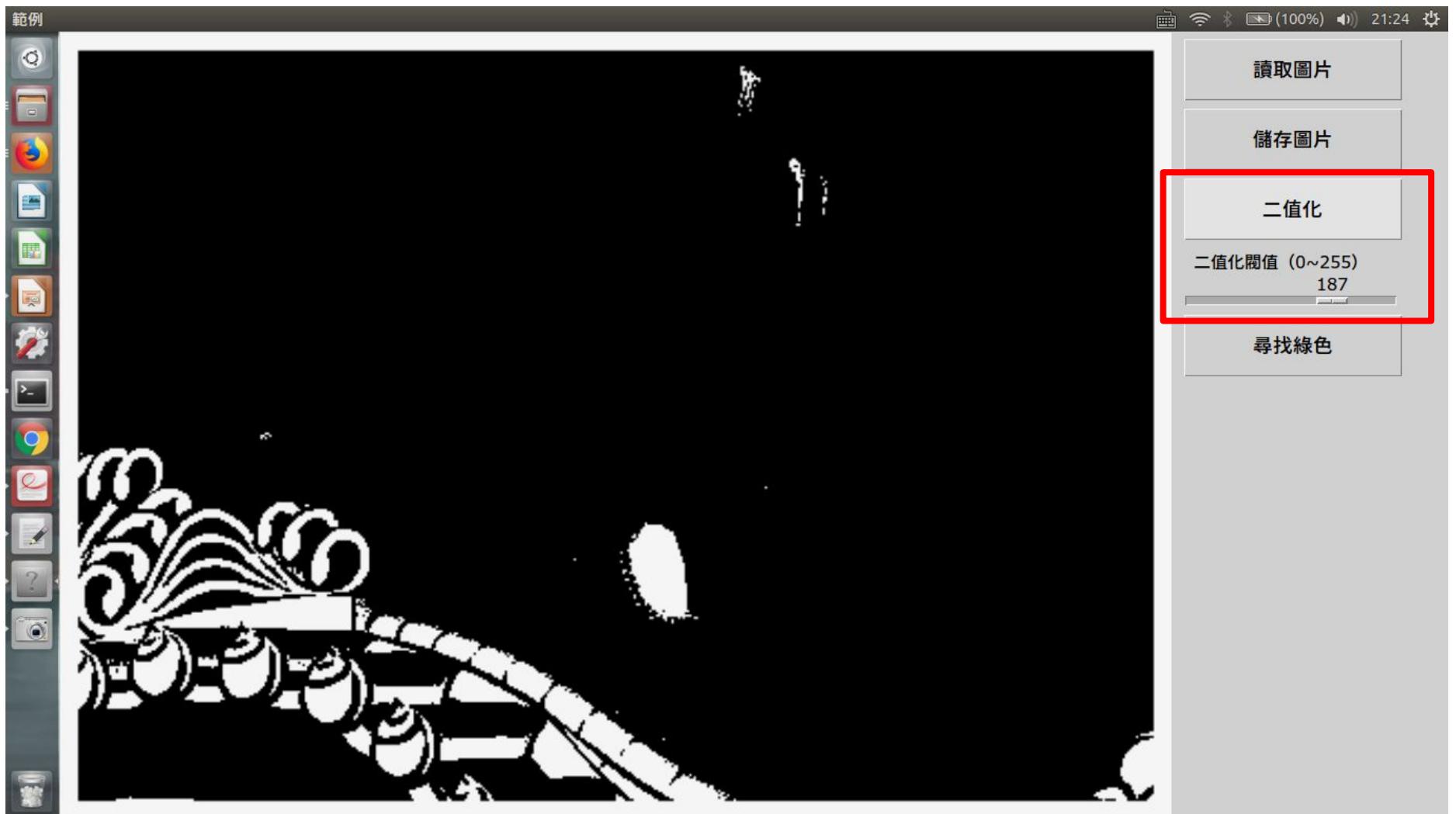
Practice-10

python opencv_FindGreen_practice.py



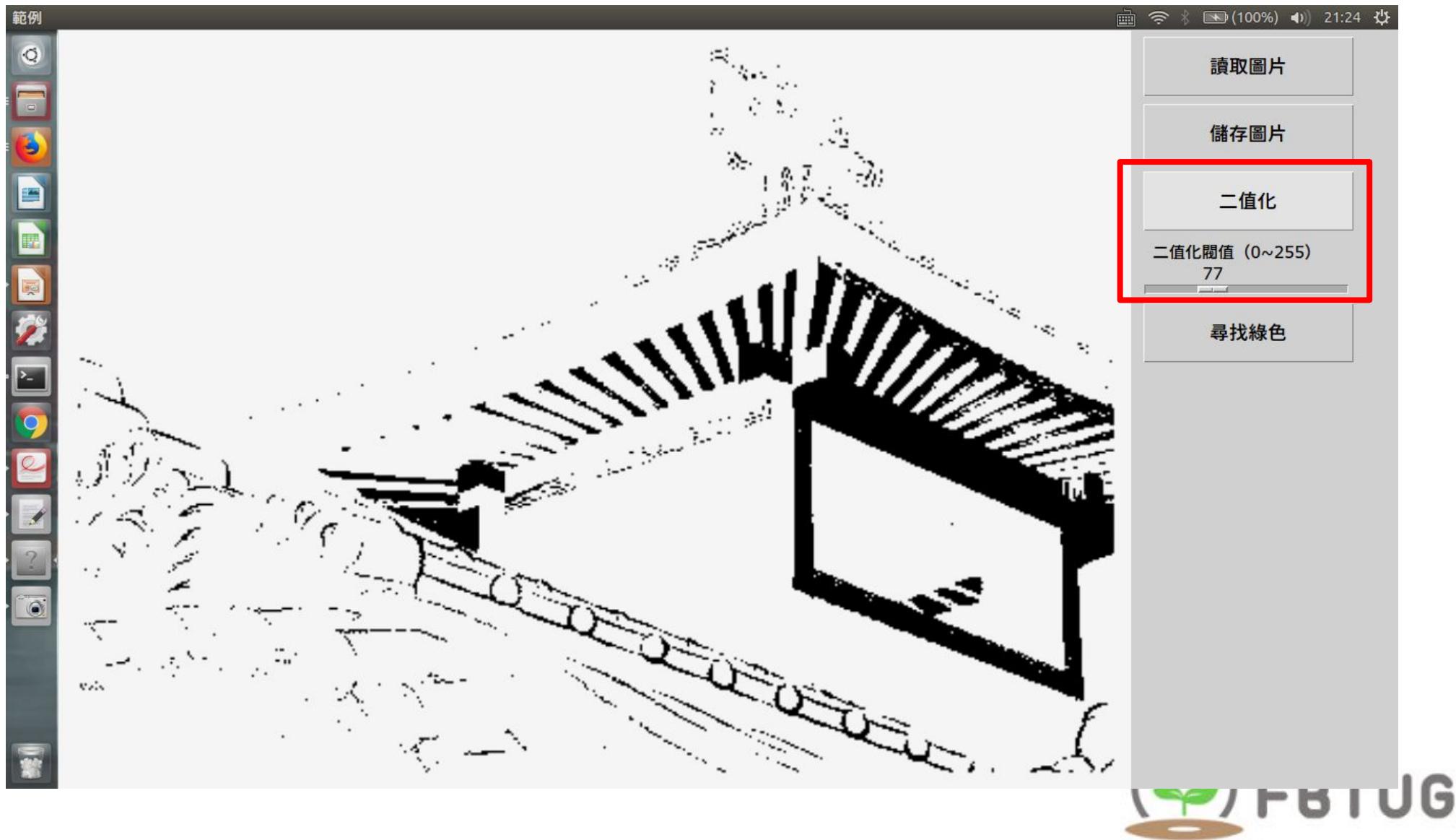
Practice-10

python opencv_FindGreen_practice.py



Practice-10

python opencv_FindGreen_practice.py



Practice-11

Add the following content after **def btn_findGreen_click(self):**:

```
124     # ===== 尋找綠色 =====
125     def btn_findGreen_click(self):
126         #---- BGR轉成 hsv通道
127         imgHSV = cv2.cvtColor(self.frame.copy(), cv2.COLOR_BGR2HSV)
128         #---- 定義上下界線 [ OpenCV (H,S,V): (0~180, 0~255, 0~255) ;
129         #           H,S,V: (0~360, 0~1, 0~1) ]
130         lower_color = np.array([38, 43, 46])
131         upper_color = np.array([83, 225, 225])
132
133         plants = cv2.inRange(imgHSV, lower_color , upper_color)
134         ctrs, _ = cv2.findContours
135         (plants, cv2.RETR_EXTERNAL ,cv2.CHAIN_APPROX_SIMPLE)
136         ctrs = filter(lambda x : cv2.contourArea(x) > 13 , ctrs)
137         image= self.frame.copy()
138         for c in ctrs:
139             cv2.drawContours(image, [c], 0, (0,0,255), 2)
140         self.display_panel(image)
141
142     pass
```

Practice-11

H: Hue, 翻譯為色相，指的是顏色的顏色屬性，如紅、黑、黃色等等，不同顏色其值不同。

	黑	灰	白	红	橙	黄	绿	青	蓝	紫	也
hmin	0	0	0	0	156	11	26	35	78	100	125
hmax	180	180	180	10	180	25	34	77	99	124	155
smin	0	0	0	43	43	43	43	43	43	43	
smax	255	43	30	255	255	255	255	255	255	255	
vmin	0	46	221	46	46	46	46	46	46	46	
vmax	46	220	255	255	255	255	255	255	255	255	

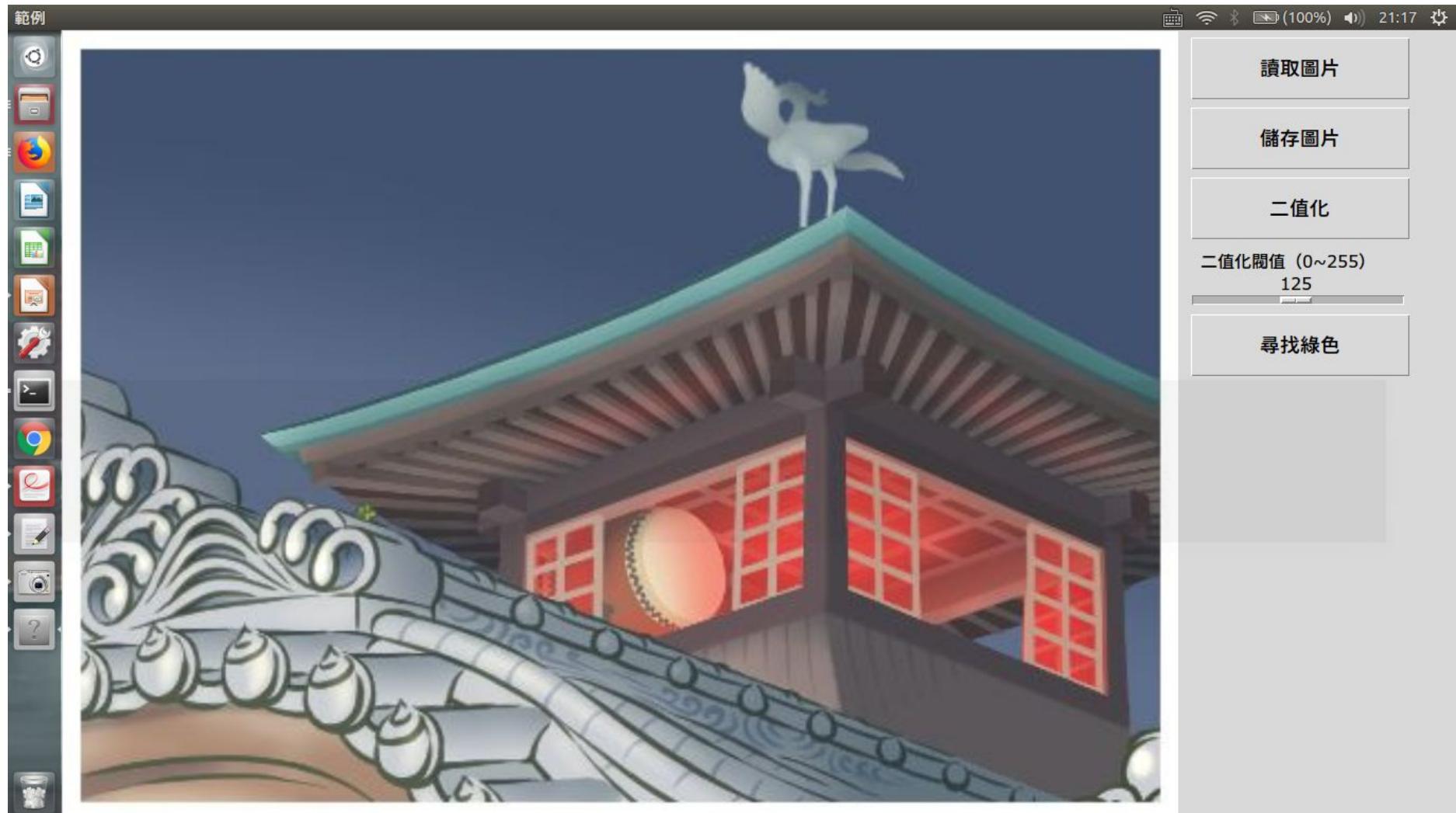
Practice-11

Add the following content after **def btn_findGreen_click(self):**:

```
124     # ===== 尋找綠色 =====
125     def btn_findGreen_click(self):
126         #---- BGR轉成 hsv通道
127         imgHSV = cv2.cvtColor(self.frame.copy(), cv2.COLOR_BGR2HSV)
128         #---- 定義上下界線 [ OpenCV (H,S,V): (0~180, 0~255, 0~255) ;
129         #           H,S,V: (0~360, 0~1, 0~1) ]
130         lower_color = np.array([38, 43, 46])
131         upper_color = np.array([83, 225, 225])
132
133         plants = cv2.inRange(imgHSV, lower_color , upper_color)
134         ctrs, _ = cv2.findContours
135         (plants, cv2.RETR_EXTERNAL ,cv2.CHAIN_APPROX_SIMPLE)
136         ctrs = filter(lambda x : cv2.contourArea(x) > 13 , ctrs)
137         image= self.frame.copy()
138         for c in ctrs:
139             cv2.drawContours(image, [c], 0, (0,0,255), 2)
140         self.display_panel(image)
141
142     pass
```

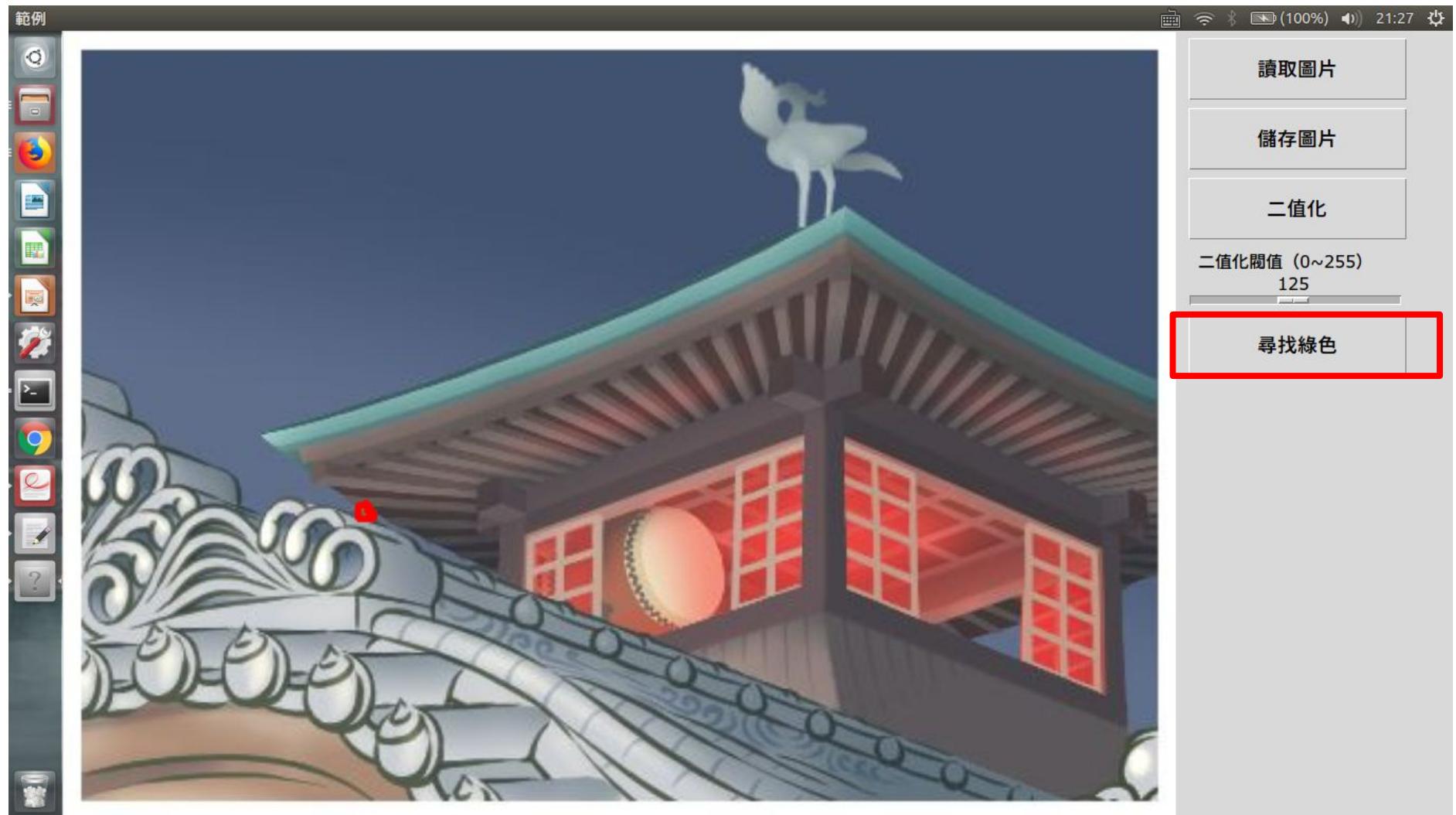
Practice-11

python opencv_FindGreen_practice.py



Practice-11

python opencv_FindGreen_practice.py



Thanks for your attention

