

快速訓練 Donkey Car

台灣樹莓派 <sosorry@raspberrypi.com.tw>
2020/05/01 @VNU

CC (Creative Commons)

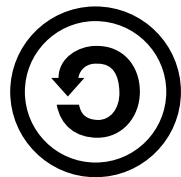
姓名標示 — 非商業性 — 相同方式分享



姓名標示 — 你必須給予 適當表彰、提供指向本授權條款的連結，以及 指出（本作品的原始版本）是否已被變更。你可以任何合理方式為前述表彰，但不得以任何方式暗示授權人為你或你的使用方式背書。



非商業性 — 你不得將本素材進行商業目的之使用。



相同方式分享 — 若你重混、轉換本素材，或依本素材建立新素材，你必須依本素材的授權條款來散布你的貢獻物。



關於我們

- Raspberry Pi 官方經銷商



about 台灣樹莓派

- 專注於 Raspberry Pi 應用與推廣
- 舉辦社群聚會 / 工作坊 / 讀書會 / 黑客松
- Website:
 - <https://www.raspberrypi.com.tw/>
- Facebook:
 - 搜尋 RaspberryPi.Taiwan
 - <https://www.facebook.com/RaspberryPi.Taiwan>



分享 x 教學

- COSCUP, MakerConf, PyCon, HKOSCon 講者
- 投影片
 - <http://www.slideshare.net/raspberrypi-tw/presentations>
- 程式碼
 - <https://github.com/raspberrypi-tw>



學習路徑



Pi選購指南



Pi設定安裝



Linux系統管理



Python程式設計

I/O硬體控制

GPIO學習套件

初

感測器學習套件
(基礎/進階)

中

空氣盒子套件
(PiM25)

初

Win10開發套件

初

智慧開關套件

初

Linux Driver
學習套件

進

無線/IoT

RFID/NFC
門禁系統

初

LoRa IoT
閘道器套件

初

生理資訊
監控IoT(藍牙)

初

毫米波人流/熱點監控
(mmWave)

初

相機/影像處理

特色相機改裝套件

初

寵物小車套件

初

自控機器手臂套件

中

小鴨車套件
(Duckietown)

進

人工智慧

驢車套件
(DonkeyCar)

初

AIY Vision Kit

中

Intel神經運算棒

中

Google Coral
USB加速器

中

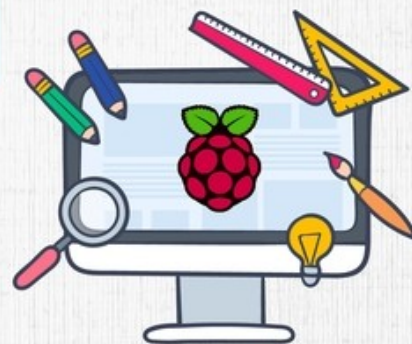
語音/訊號處理

智慧音箱套件

初

AIY Voice Kit

初

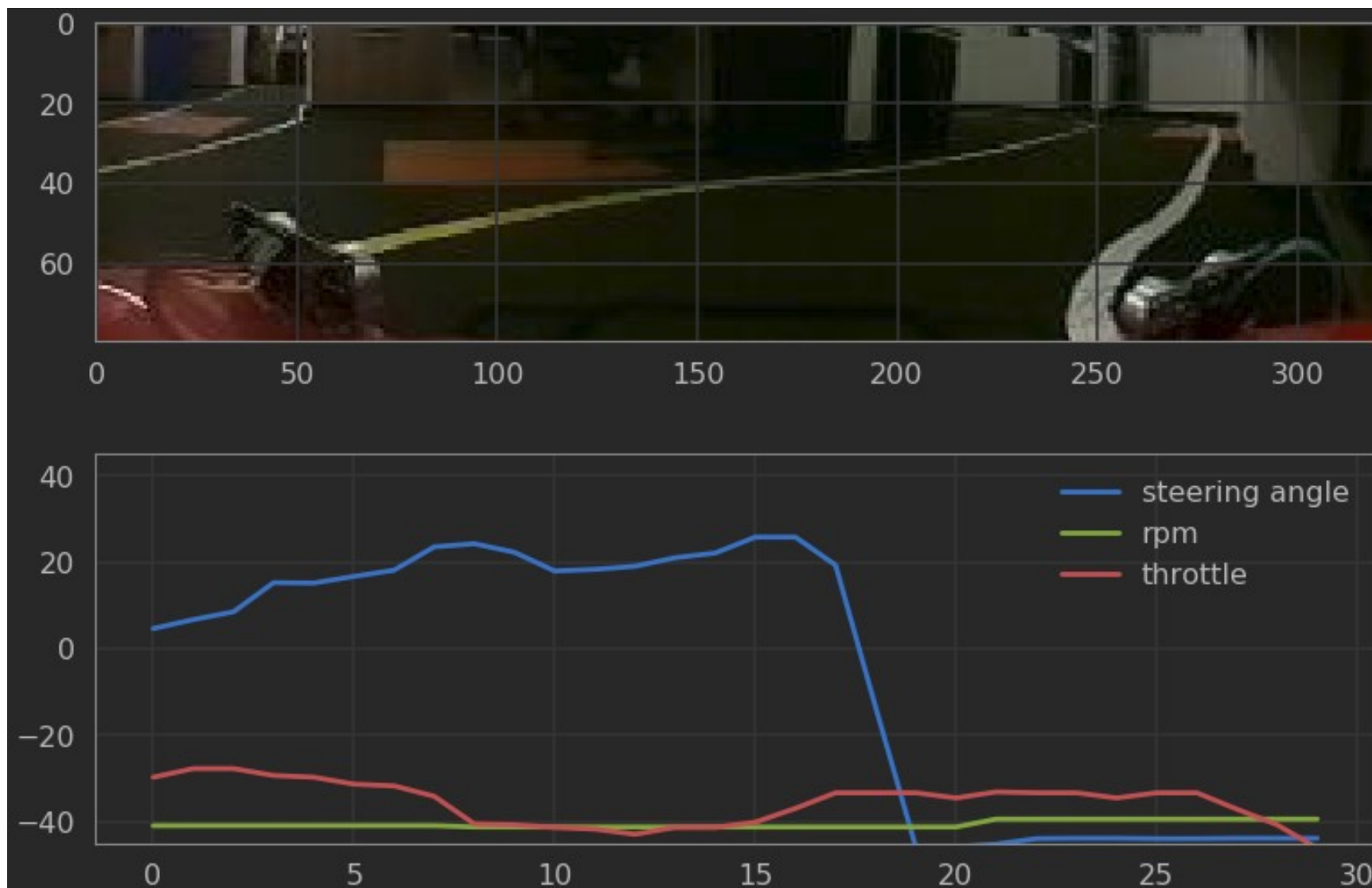


初 初階課程

中 中階課程

進 進階課程

目標：根據影像預測油門和轉向






<https://www.donkeycar.com/updates>

實做步驟

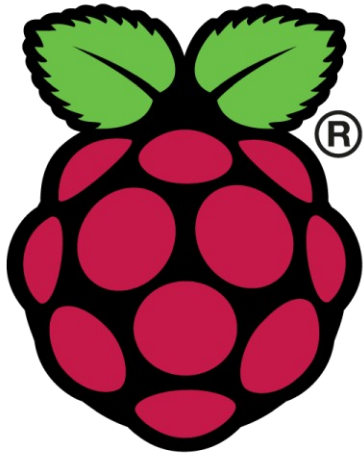
- 根據收集到的影像資料進行訓練，將得到模型
- Donkey Car 根據模型進行自走

角色

-  • Raspberry Pi: 負責收集影像，根據模型自走
-  • Google Colab: 負責訓練模型
-  • Github: 資料交換中心

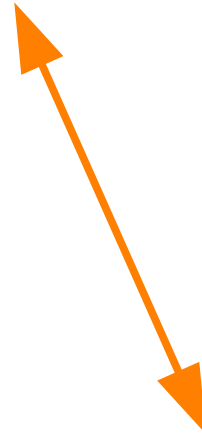
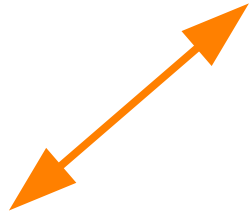


GitHub



Raspberry Pi

colab



git 和 github

- git 是版本控制系統 , github 是版本控制服務
- git 基本指令
 - # 在本地端做版本控制
 - git init
 - git add .
 - git commit -m " 喜歡的註解 "
 - # 和遠端 Repository 互動
 - git remote add origin https://github.com/<ID>/<REPO>
 - git push -u origin master
 - git pull

Google Colab

- Jupyter Notebook + Markdown 文件
- 免費使用 GPU/TPU(每天 GPU 可使用 12hrs)

操作步驟

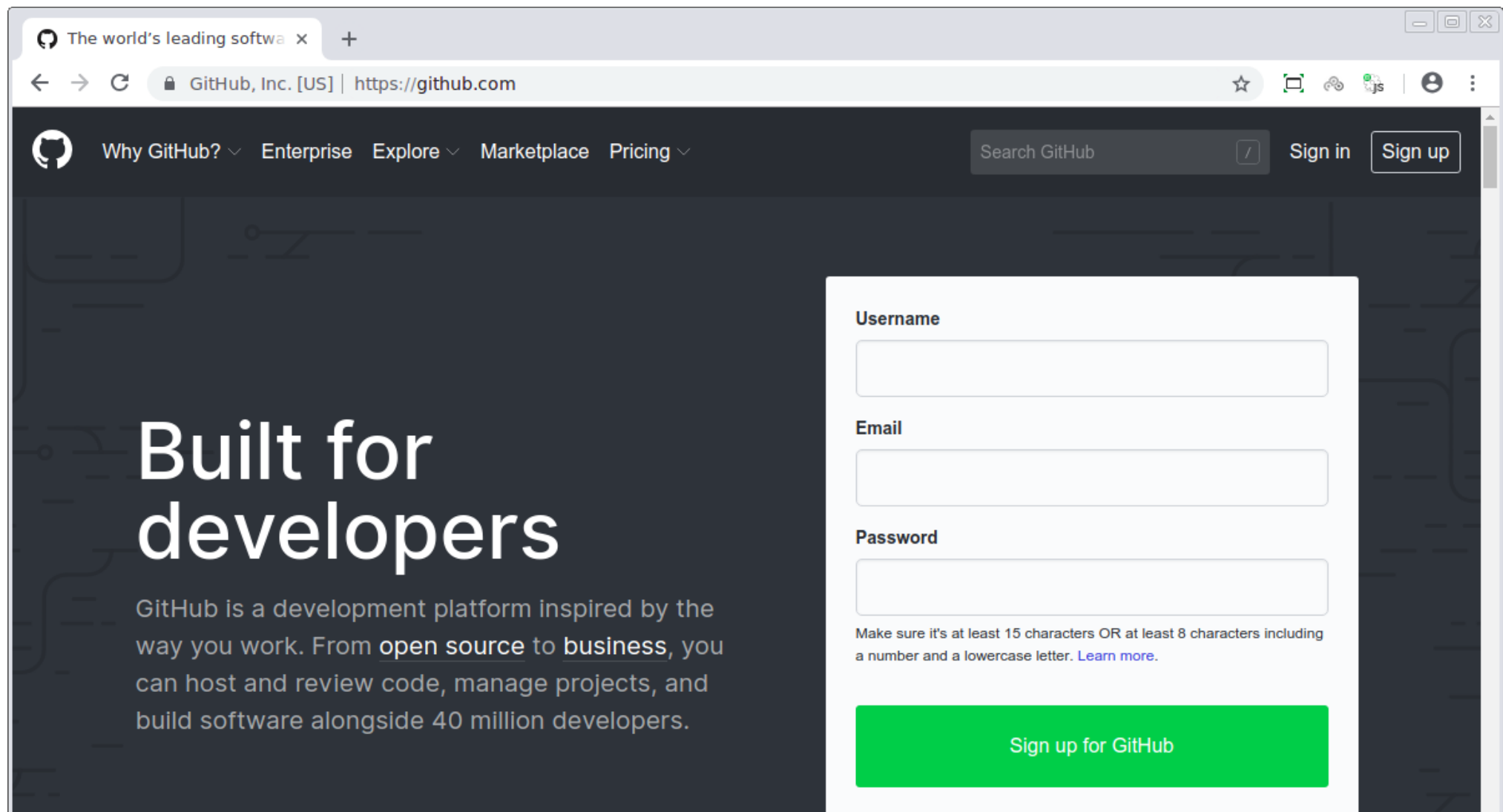
- 🐙 • 實驗 1：在 github 建立 repo
- 🍷 • 實驗 2：使用搖桿控制，取得影像和搖桿資訊
- 🍷 • 實驗 3：將影像和搖桿資訊 push 到 github
- 🏠 • 實驗 4：建立 Donkey Car 環境
- 🏠 • 實驗 5：從 github 下載影像和搖桿資訊
- 🏠 • 實驗 6：使用 CNN 或是 RNN 訓練後得到模型
- 🍷 • 實驗 7：下載模型並執行自走功能

實驗 1: 在 github 建立 repo

目的：建立資料交換中心

建立 github 帳號

- `https://github.com`



The screenshot shows the GitHub homepage with a dark background. On the left, the text "Built for developers" is prominently displayed. Below it, a paragraph describes GitHub as a development platform. On the right, a white sign-up form is overlaid. The form has three input fields: "Username", "Email", and "Password". Below the "Password" field, there is a note about password requirements and a link to "Learn more". At the bottom of the form is a large green button labeled "Sign up for GitHub". The browser's address bar shows "https://github.com" and the page title is "The world's leading software development platform".

The world's leading software development platform

GitHub, Inc. [US] | <https://github.com>

Why GitHub? ▾ Enterprise Explore ▾ Marketplace Pricing ▾

Search GitHub / Sign in Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 40 million developers.

Username

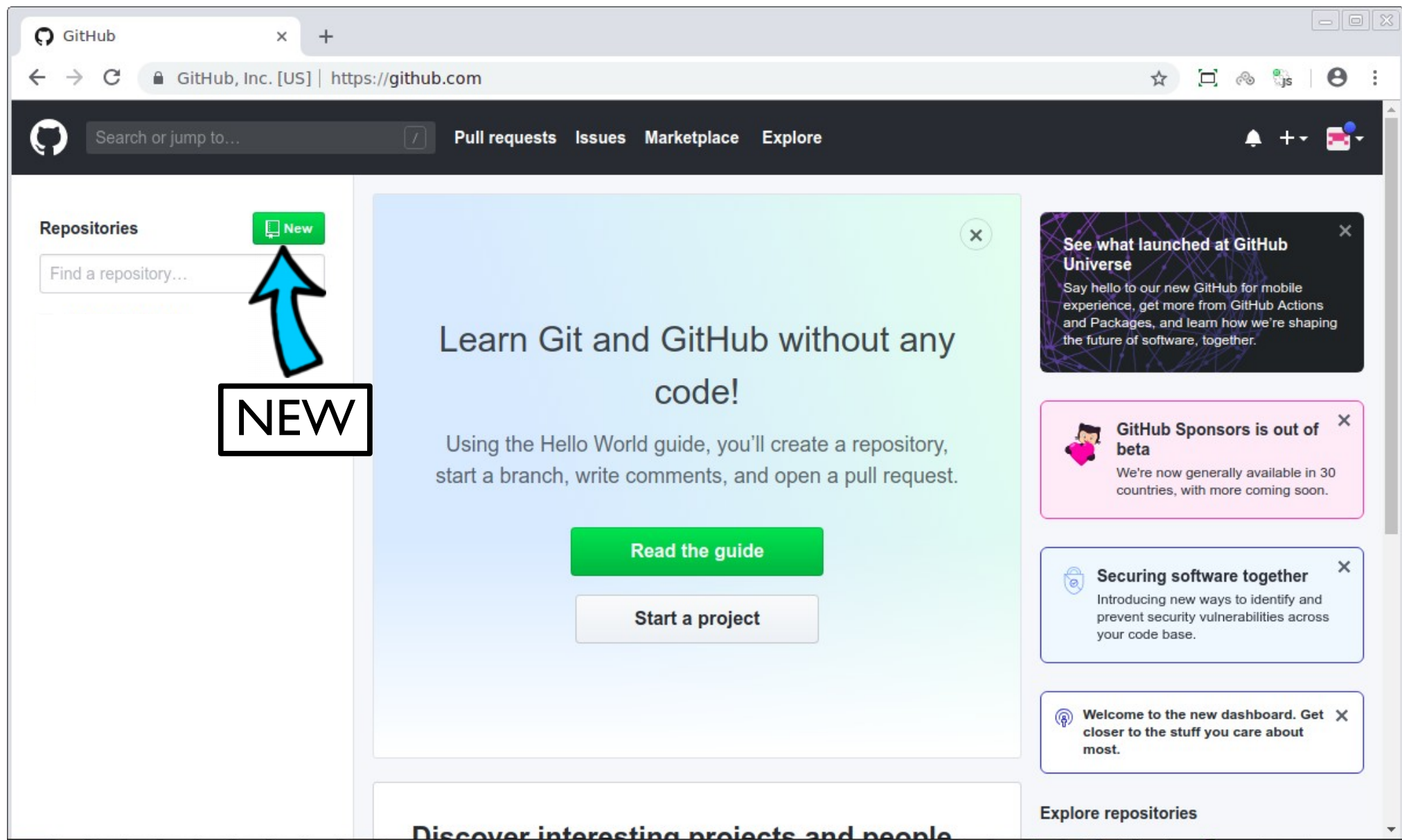
Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

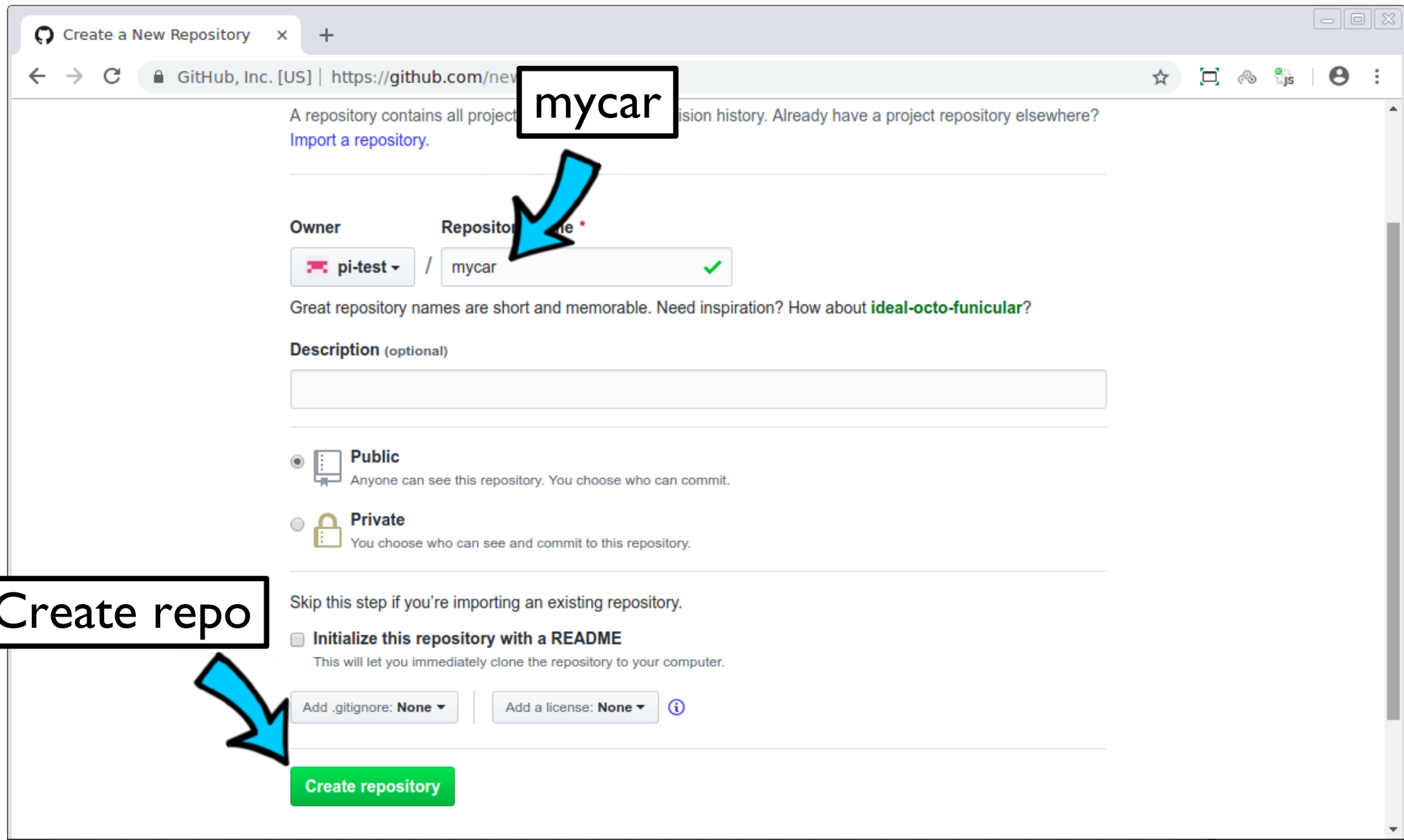
Sign up for GitHub

建立新的 repo



The screenshot shows the GitHub homepage in a web browser. The browser's address bar displays "GitHub, Inc. [US] | https://github.com". The top navigation bar includes a search bar, "Pull requests", "Issues", "Marketplace", and "Explore". On the left sidebar, under the "Repositories" section, there is a "New" button with a plus icon. A blue arrow points from a black-bordered box containing the word "NEW" to this button. The main content area features a large light blue box with the text "Learn Git and GitHub without any code!" and a subtext "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." Below this text are two buttons: a green "Read the guide" button and a grey "Start a project" button. The right sidebar contains several promotional cards, including "See what launched at GitHub Universe", "GitHub Sponsors is out of beta", "Securing software together", and "Welcome to the new dashboard. Get closer to the stuff you care about most."

為 repo 命名 (例如 mycar)



The screenshot shows the GitHub 'Create a New Repository' page. A blue arrow points from a box labeled 'mycar' to the 'Repository name' field, which contains 'mycar' with a green checkmark. Another blue arrow points from a box labeled 'Create repo' to the 'Create repository' button at the bottom. The page includes fields for Owner (pi-test), Repository name (mycar), Description (optional), and visibility options (Public and Private). It also has checkboxes for 'Initialize this repository with a README' and dropdowns for 'Add .gitignore' and 'Add a license'.

Create a New Repository

GitHub, Inc. [US] | <https://github.com/new>

A repository contains all project files and revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner: pi-test / Repository name: mycar ✓

Great repository names are short and memorable. Need inspiration? How about [ideal-octo-funicular?](#)

Description (optional)

☒ **Public**
Anyone can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None ⓘ


Create repository

實驗 2: 用搖桿控制，取得影像搖桿資訊

目的：學習前要先收資料

使用搖桿控制

- 用範例執行 (`sample_manage.py`)
- (env)\$ `cd ~/mycar`
- (env)\$ `cp ~/donkeypart_ps3_controller/sample_manage.py .`
- (env)\$ `python3 sample_manage.py drive --js`



有一個點

注意事項

- 先確定搖桿是否配對成功？
- 左右控制不會拍照（收資料），前後才會
- 搖桿控制是數位資料，比例從 `myconfig.py` 修改
- 每次重新執行程式，資料會紀錄在 `data` 目錄下
- 照片最少收 5000 張

2.4G 搖桿



電源開關

測試搖桿和 Donkey Car 的移動行為





實驗 3: 將資訊 push 到 github

目的：使用 github 進行資料交換

設定 git 帳號 (需自行設定)

```
$ cd ~/mycar
```

```
$ git config --global user.email "you@example.com"
```

```
$ git config --global user.name "Your Name"
```



將紅字換成自己的資訊
記得要加雙引號

- 如果編輯器習慣用 VIM 可以再加一行

```
$ git config --global core.editor "vim"
```

將資訊 push 到 github

```
$ cd ~/mycar
```

```
$ git init
```

```
$ git add .
```

```
$ git commit -m " 喜歡的註解 "
```

```
$ git remote add origin
```

```
https://github.com/<ID>/<REPO>
```

```
$ git push -u origin master
```

 之後要再 push 到 github

```
$ cd ~/mycar
```

```
$ git init
```

```
$ git add .
```

```
$ git commit -m " 喜歡的註解 "
```

```
$ git remote add origin
```

```
https://github.com/<ID>/<REPO>
```

```
$ git push -u origin master
```

實驗 4: 建立 Donkey Car 環境

目的：使用 Google Colab 訓練

Google Colab

- 先將以下連結建立副本 (File > Save a copy in drive)
- <https://colab.research.google.com/drive/1h-1CNL9usSdsW6TNvejUTsNBYhbnS5F>
- 短網址 : <http://bit.ly/20DIpNw>

實驗 5: 從 github 下載影像和搖桿資訊

目的：根據 Colab 操作

實驗 6: 使用 CNN 或是 RNN 訓練

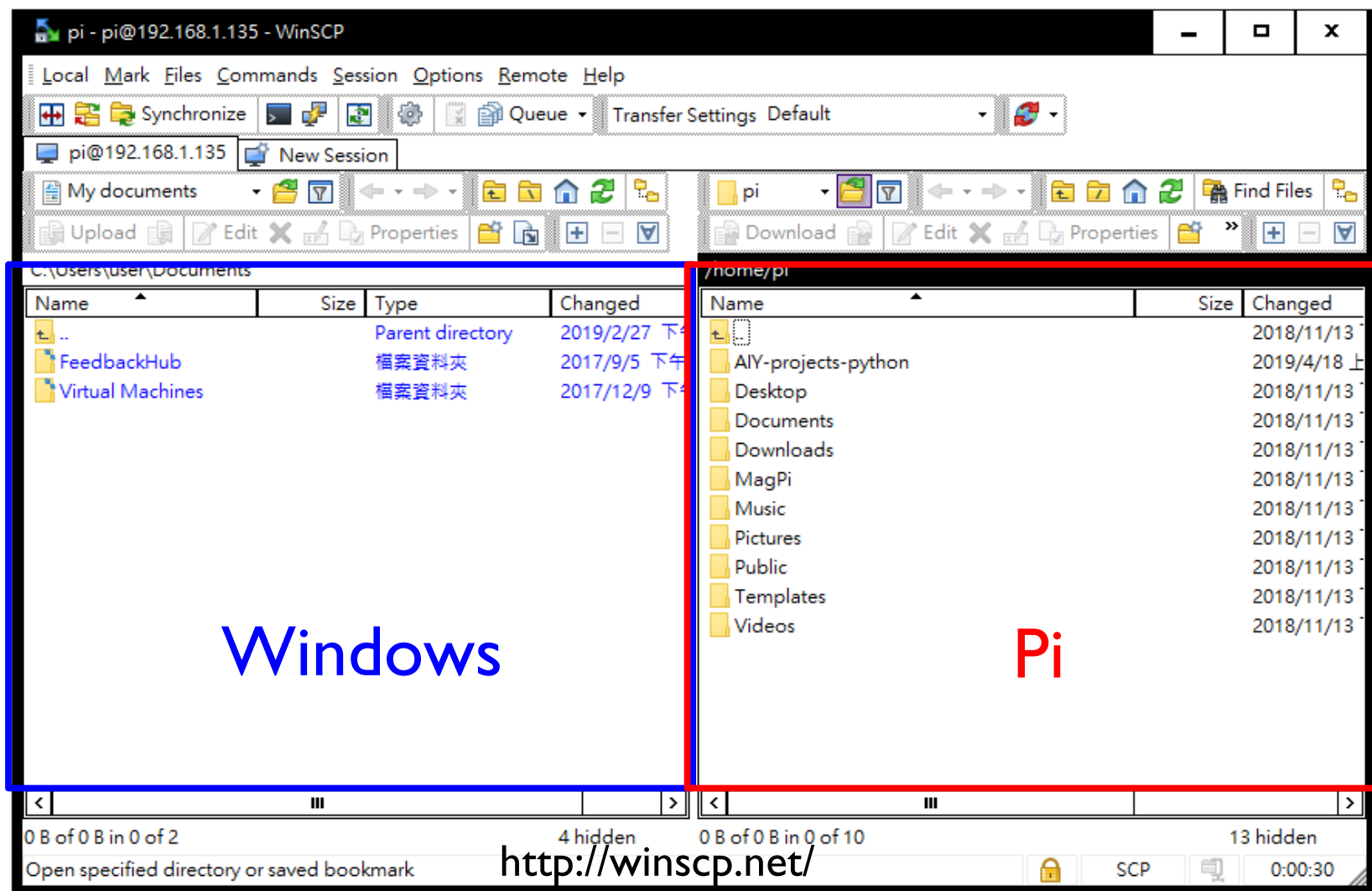
目的：觀察不同模型的結果

實驗 7: 下載模型並執行自走功能

目的：實際驗證

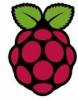
使用 SCP 將模型上傳到 Pi

- 左右兩邊都可以做檔案傳輸



執行自走功能

- `(env)$ cd ~/mycar`
- `(env)$ python3 sample_manage.py drive
--model models/lin_1.h5`
- 訊息將會提示開啟網頁 `<ipaddress:8887>`



<ipaddress:8887>

Control Mode ⓘ

Joystick

Gamepad

Device Tilt

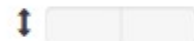
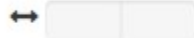
Max Throttle

Select Max Throttle ▼

Throttle Mode

User ▼

Angle &
Throttle



Mode & Pilot

User (d) ▼

Start Vehicle



Click/touch to use joystick.

選擇 angle

Start Vehicle

慢慢按鍵盤的 i，車子逐漸前進並自走。按空白鍵停止

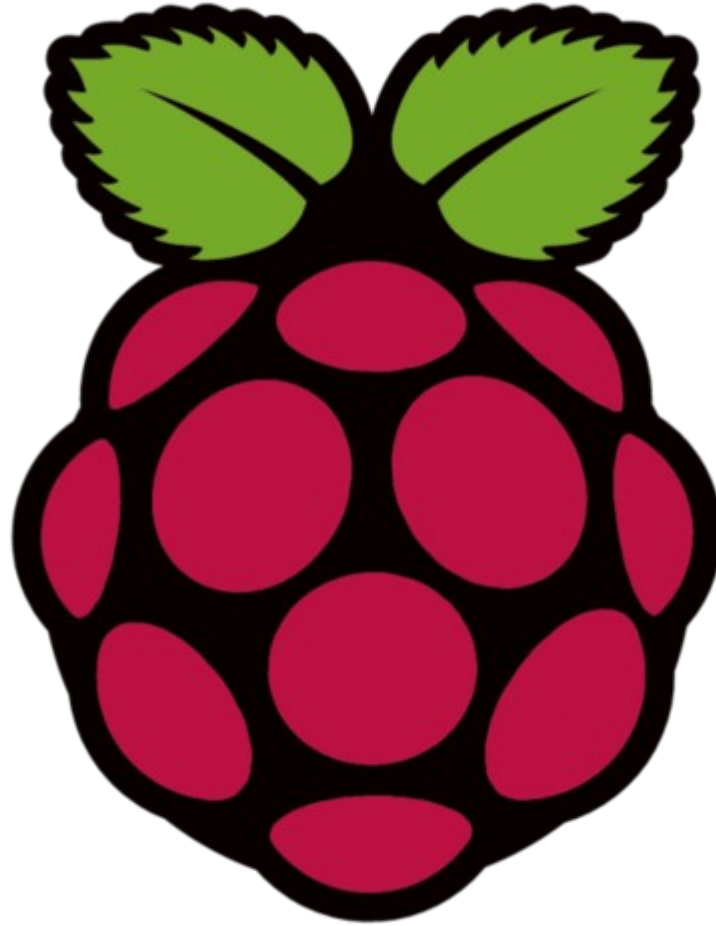
Keras 神經網路模型

- 可用神經網路模型 (預設 linear)
- `--type=linear|categorical|rnn|latent`
- linear: 平穩, 無上下限, 可能會暴衝
- categorical: 平穩且適合油門, 但有上下限
- rnn: 適合連續資料, 但訓練資料時間長
- latent: 可將模擬器所學應用到新場景

指定 RNN 模型注意事項

- 訓練時，需指定 type 為 rnn
- `$ python3 sample_manage.py train --model models/rnn_1.h5 --type rnn`
- 推論時，也需指定 type 為 rnn
- `$ python3 sample_manage.py drive --model models/rnn_1.h5 --type rnn`

Raspberry Pi Rocks the World



Thanks