111-1自主學習

用Python開發星座辨識定向系統

217 葉偉權、劉育誠、黃禎鈺

```python
class selfDirectedLearning:
    def __init__(self, learners) -> None:
        self.learner1 = learners[0]
        self.learner2 = learners[1]
        self.learner3 = learners[2]
    def __str__(self) -> str:
        return f"{self.learner1} and {self.learner2} and {self.learner3}"

    def endReport(self):
        return "Please continue watching... "

if __name__ == "__main__":
    fightersOf217 = selfDirectedLearning(["葉偉權", "劉育誠", "黃禎鈺"])
    print(f"We are {fightersOf217}.", end=" ")
    print(fightersOf217.midTermReport())
```

動機與目的

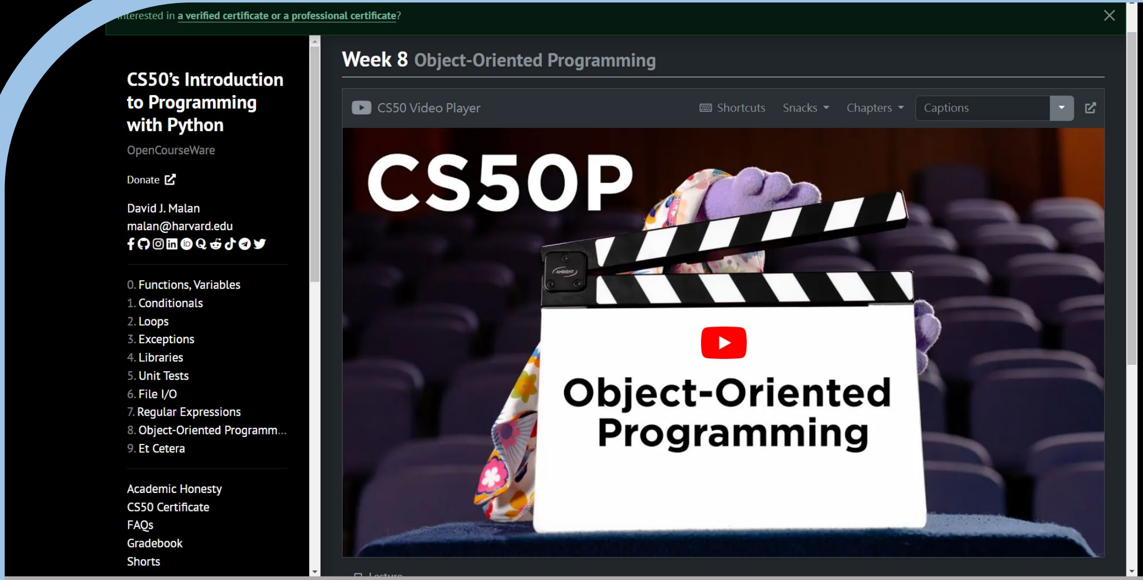
動機

目前較有名的導星軟體StarWalk2是以陀螺儀和加速度計判斷星星位置,再以AR模式顯示在鏡頭之上,但此方式較不精準。因此我們希望研發一款可以用手機相機拍攝星空便能辨識星座圖樣的導星軟體。

目的

學習Python語法，並開發以Tensorflow為基礎的卷積神經網路，訓練辨識黃道十二宮的星座

Python學習



nterested in a verified certificate or a professional certificate?

CS50's Introduction to Programming with Python

OpenCourseWare

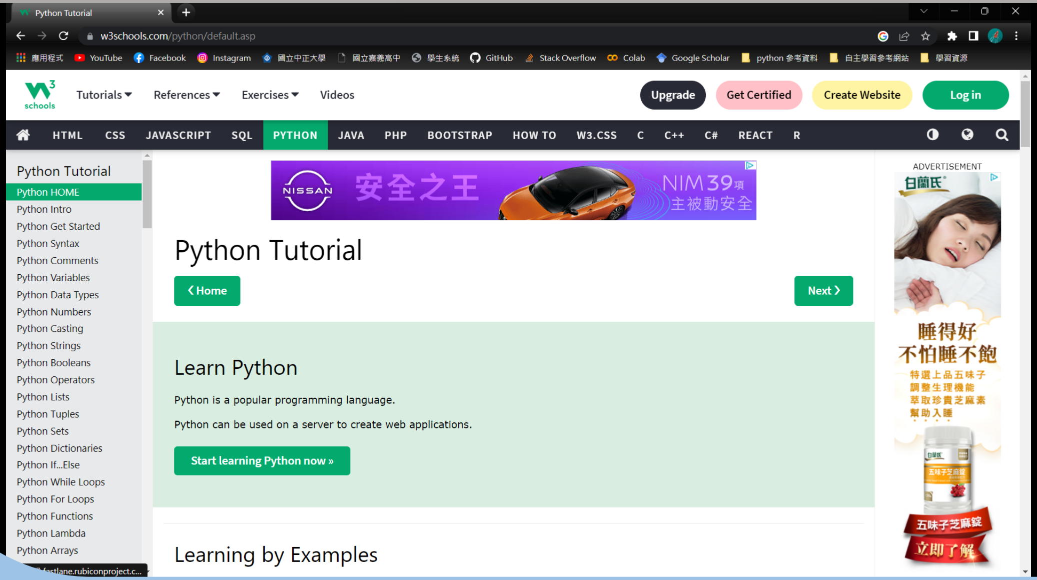
Donate

David J. Malan
malan@harvard.edu

0. Functions, Variables
1. Conditionals
2. Loops
3. Exceptions
4. Libraries
5. Unit Tests
6. File I/O
7. Regular Expressions
8. Object-Oriented Programm...
9. Et Cetera

Academic Honesty
CS50 Certificate
FAQs
Gradebook
Shorts

Week 8 Object-Oriented Programming



Python Tutorial

- Python HOME
- Python Intro
- Python Get Started
- Python Syntax
- Python Comments
- Python Variables
- Python Data Types
- Python Numbers
- Python Casting
- Python Strings
- Python Booleans
- Python Operators
- Python Lists
- Python Tuples
- Python Sets
- Python Dictionaries
- Python If...Else
- Python While Loops
- Python For Loops
- Python Functions
- Python Lambda
- Python Arrays

Python Tutorial

❮ Home　　Next ❯

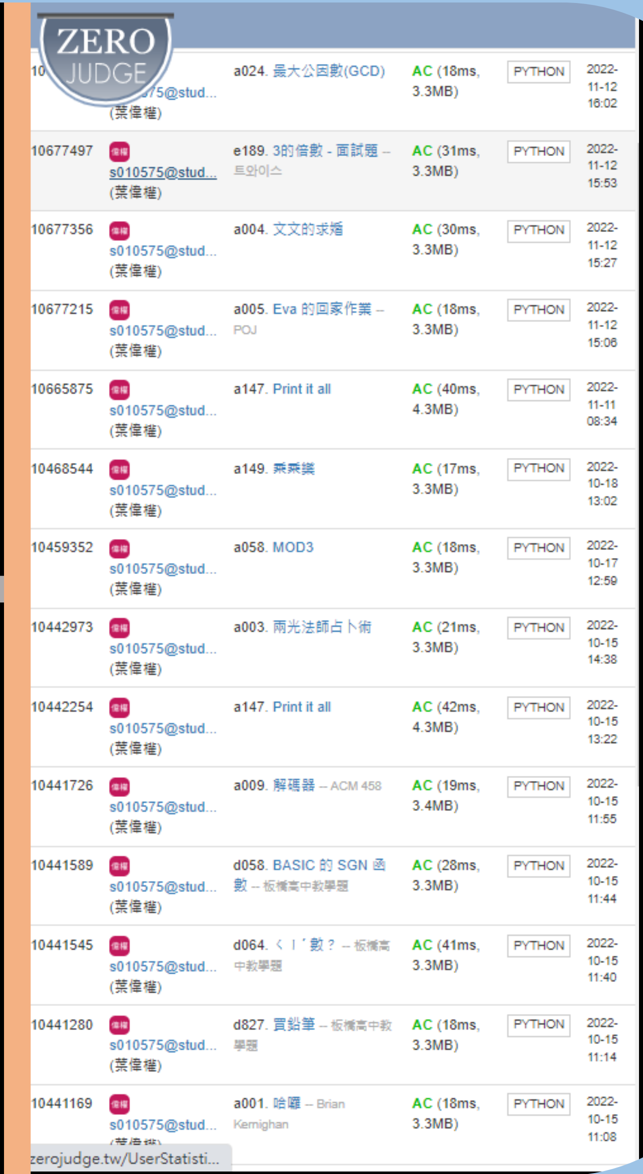
Learn Python

Python is a popular programming language.

Python can be used on a server to create web applications.

Start learning Python now »

Learning by Examples



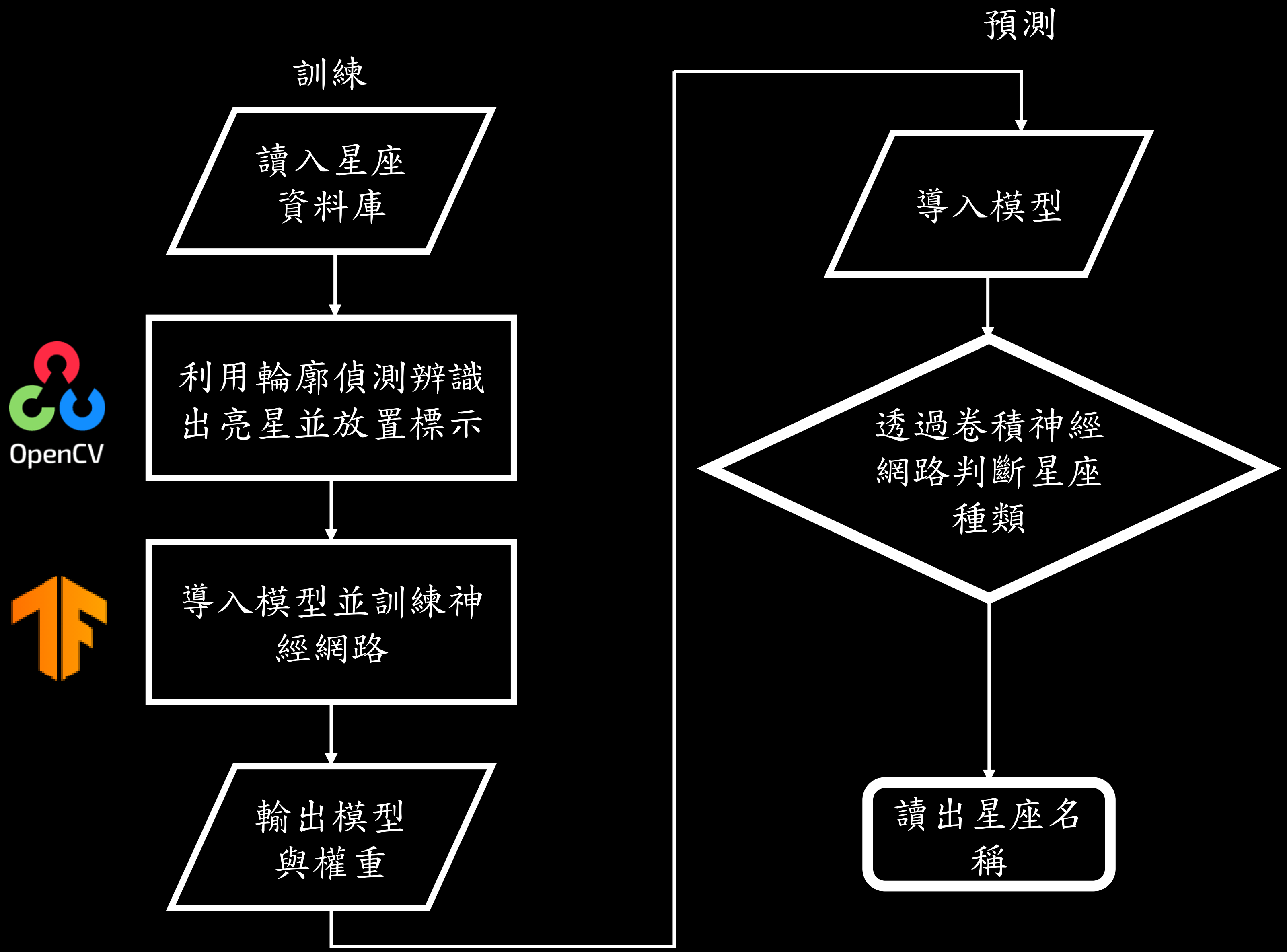
10	75@stud... (葉偉權)	a024. 最大公因數(GCD)	AC (18ms, 3.3MB)	PYTHON	2022-11-12 16:02
10677497	s010575@stud... (葉偉權)	e189. 3的倍數 - 面試題 -- 트와이스	AC (31ms, 3.3MB)	PYTHON	2022-11-12 15:53
10677356	s010575@stud... (葉偉權)	a004. 文文的求婚	AC (30ms, 3.3MB)	PYTHON	2022-11-12 15:27
10677215	s010575@stud... (葉偉權)	a005. Eva 的回家作業 -- POJ	AC (18ms, 3.3MB)	PYTHON	2022-11-12 15:06
10665875	s010575@stud... (葉偉權)	a147. Print it all	AC (40ms, 4.3MB)	PYTHON	2022-11-11 08:34
10468544	s010575@stud... (葉偉權)	a149. 乘乘樂	AC (17ms, 3.3MB)	PYTHON	2022-10-18 13:02
10459352	s010575@stud... (葉偉權)	a058. MOD3	AC (18ms, 3.3MB)	PYTHON	2022-10-17 12:59
10442973	s010575@stud... (葉偉權)	a003. 兩光法師占卜術	AC (21ms, 3.3MB)	PYTHON	2022-10-15 14:38
10442254	s010575@stud... (葉偉權)	a147. Print it all	AC (42ms, 4.3MB)	PYTHON	2022-10-15 13:22
10441726	s010575@stud... (葉偉權)	a009. 解碼器 -- ACM 458	AC (19ms, 3.4MB)	PYTHON	2022-10-15 11:55
10441589	s010575@stud... (葉偉權)	d058. BASIC 的 SGN 函數 -- 板橋高中教學題	AC (28ms, 3.3MB)	PYTHON	2022-10-15 11:44
10441545	s010575@stud... (葉偉權)	d064. ㄑㄧˊ數？ -- 板橋高中教學題	AC (41ms, 3.3MB)	PYTHON	2022-10-15 11:40
10441280	s010575@stud... (葉偉權)	d827. 買鉛筆 -- 板橋高中教學題	AC (18ms, 3.3MB)	PYTHON	2022-10-15 11:14
10441169	s010575@stud... (葉偉權)	a001. 哈囉 -- Brian Kernighan	AC (18ms, 3.3MB)	PYTHON	2022-10-15 11:08

zerojudge.tw/UserStatisti...

在一開始學習python的過程中，我們對這個語言非常陌生，因此常常感覺到挫折，不知所措，不過我們在之後開始接觸不同的教材，像是W3School、CS50P、Zerojudge...等，開始越來越熟悉此語言，且能夠慢慢活用在本次的專題中。

流程

```
#key imports
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv
import copy
import os
```



訓練

讀入星座資料庫

利用輪廓偵測辨識出亮星並放置標示

導入模型並訓練神經網路

輸出模型與權重

預測

導入模型

透過卷積神經網路判斷星座種類

讀出星座名稱

OpenCV 邊緣偵測

```python
working_directory = "/content/drive/MyDrive/R/Virgo“

blank = np.zeros((1080, 1920, 1), dtype=np.uint8)

class contourFinder:
    def __init__(self) -> None:
        cnt = 0
        self.__images = self.__imageReader()
        for image in self.__images:
            self.currimg = image
            try:
                cnt += 1
                dir = "/content/drive/MyDrive/Train/Virgo/processed-" + str(cnt) + ".png"
                self.__preProcessing(image)
                self.findContours(100, 300)
                cv.imwrite(dir, self.clone)
            except ZeroDivisionError:
                print("error")
                continue
            except KeyboardInterrupt: break

    def __imageReader(self, folder = working_directory):
        images = []
        for filename in os.listdir(folder):
            img = cv.imread(os.path.join(folder, filename))
            if img is not None: images.append(img)
        return images
```

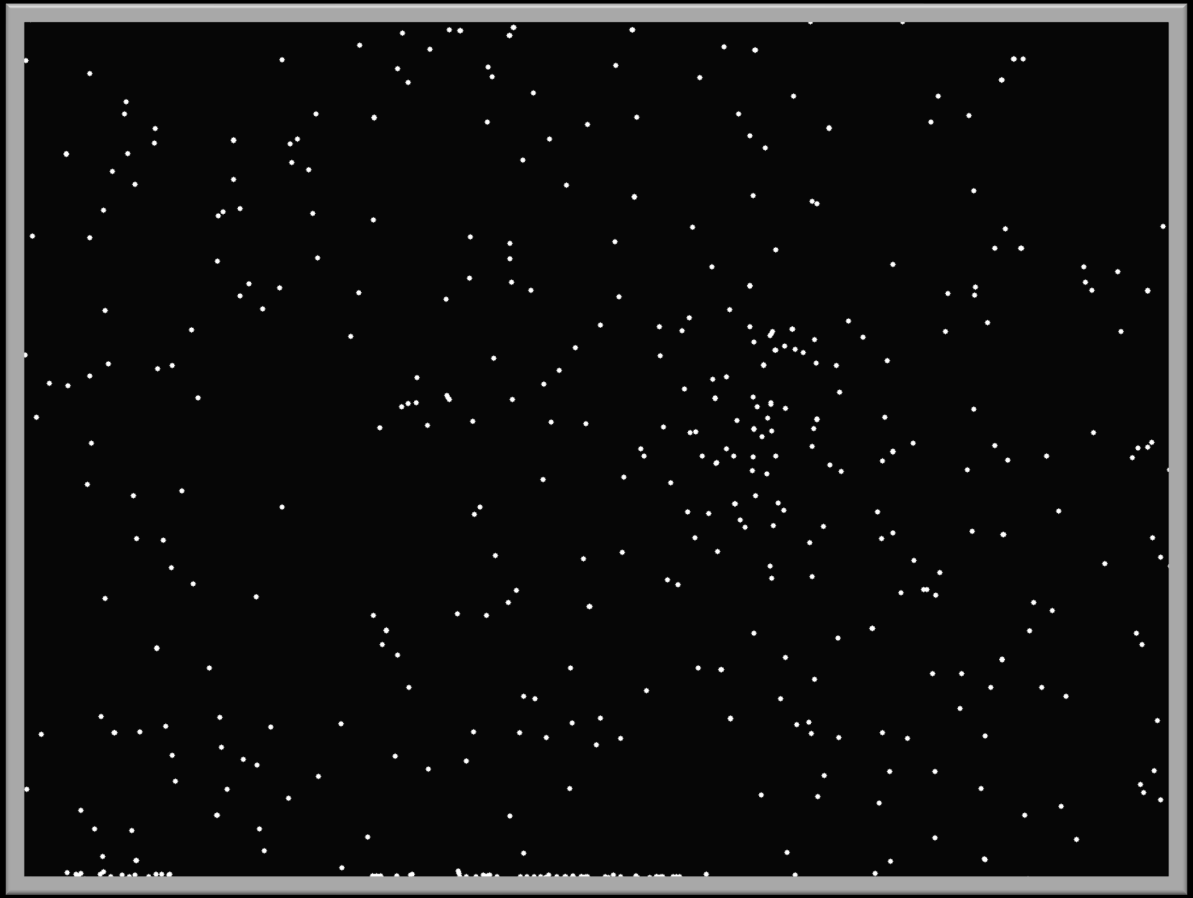
匯入照片

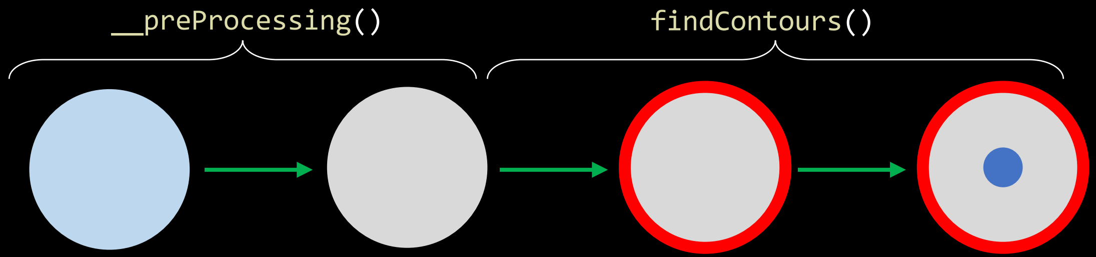
```python
    def __preProcessing(self, image):
        self.__gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
        self.__blurred = cv.GaussianBlur(self.__gray, (5, 5), 1)
```

詼階、模糊

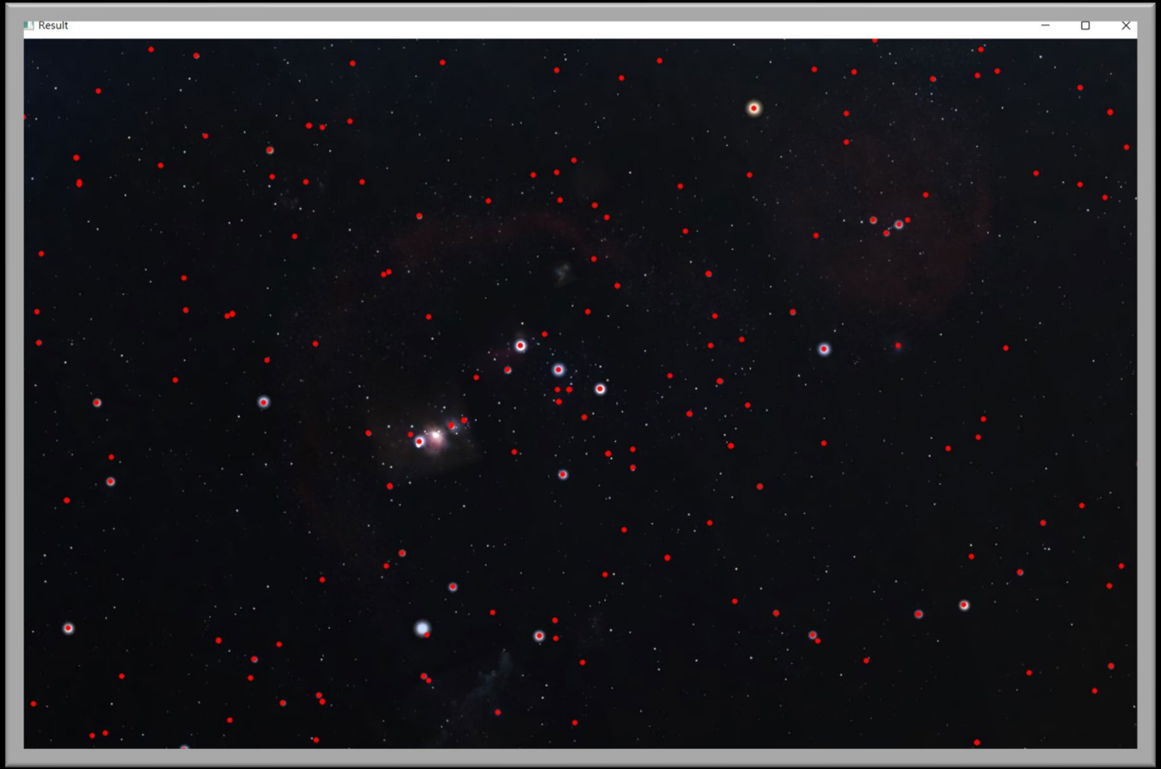
```python
    def findContours(self, t1, t2):
        tmp = self.__blurred

        self.__canneyed = cv.Canny(tmp, t1, t2)
        (self.cnts, _) = cv.findContours(self.__canneyed, cv.RETR_LIST, cv.CHAIN_APPROX_SIMPLE)
        self.clone = copy.deepcopy(blank)

        for c in self.cnts:
            M = cv.moments(c)

            if M["m00"] != 0:
                cX, cY = (int(M["m10"]/M["m00"]), int(M["m01"]/M["m00"]))
            else: cX, cY = 0, 0

            cv.circle(self.clone, (cX, cY), 3, (255, 255, 255), -1)

image = contourFinder()
```

提取邊緣
點出中心

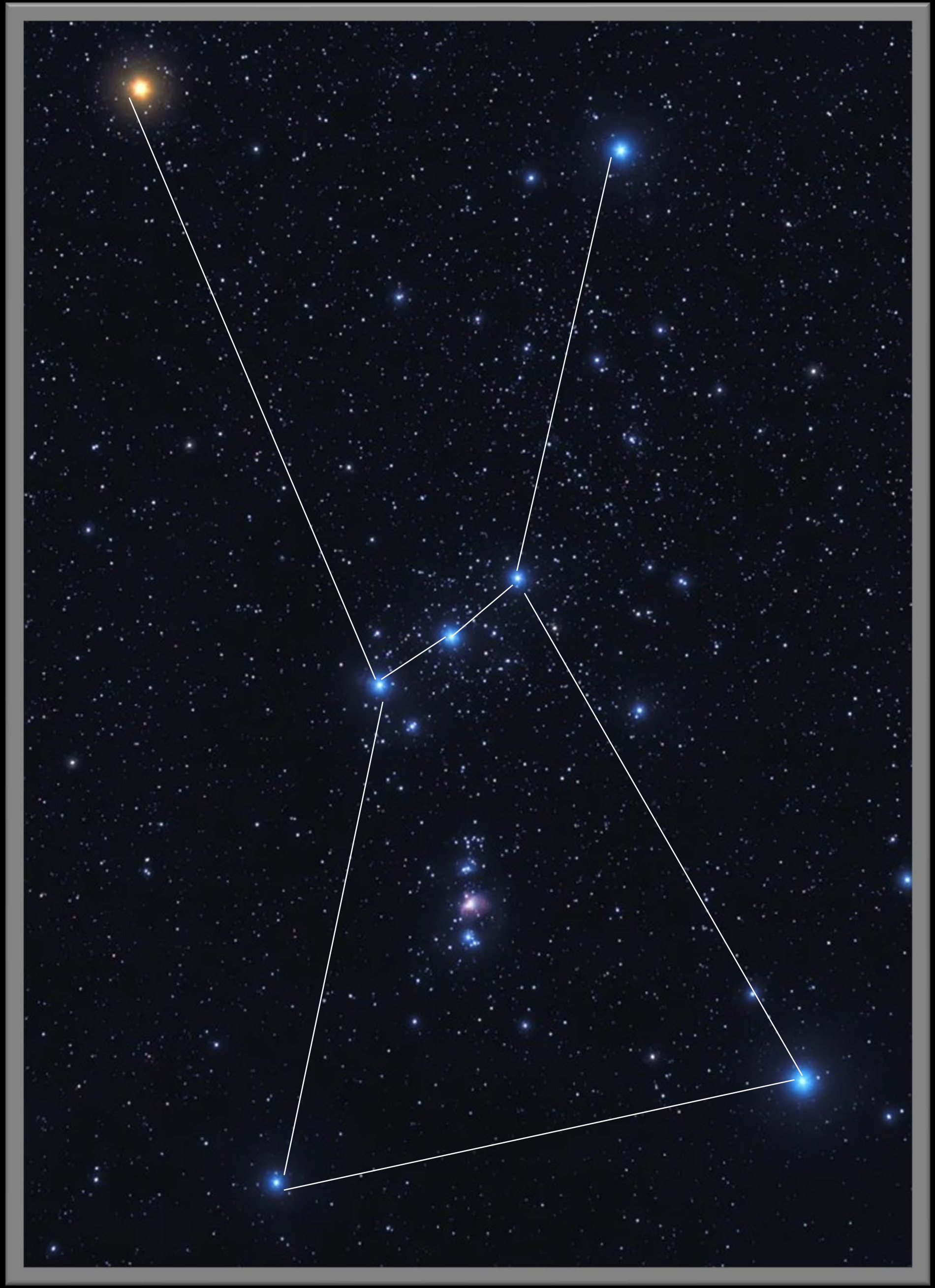


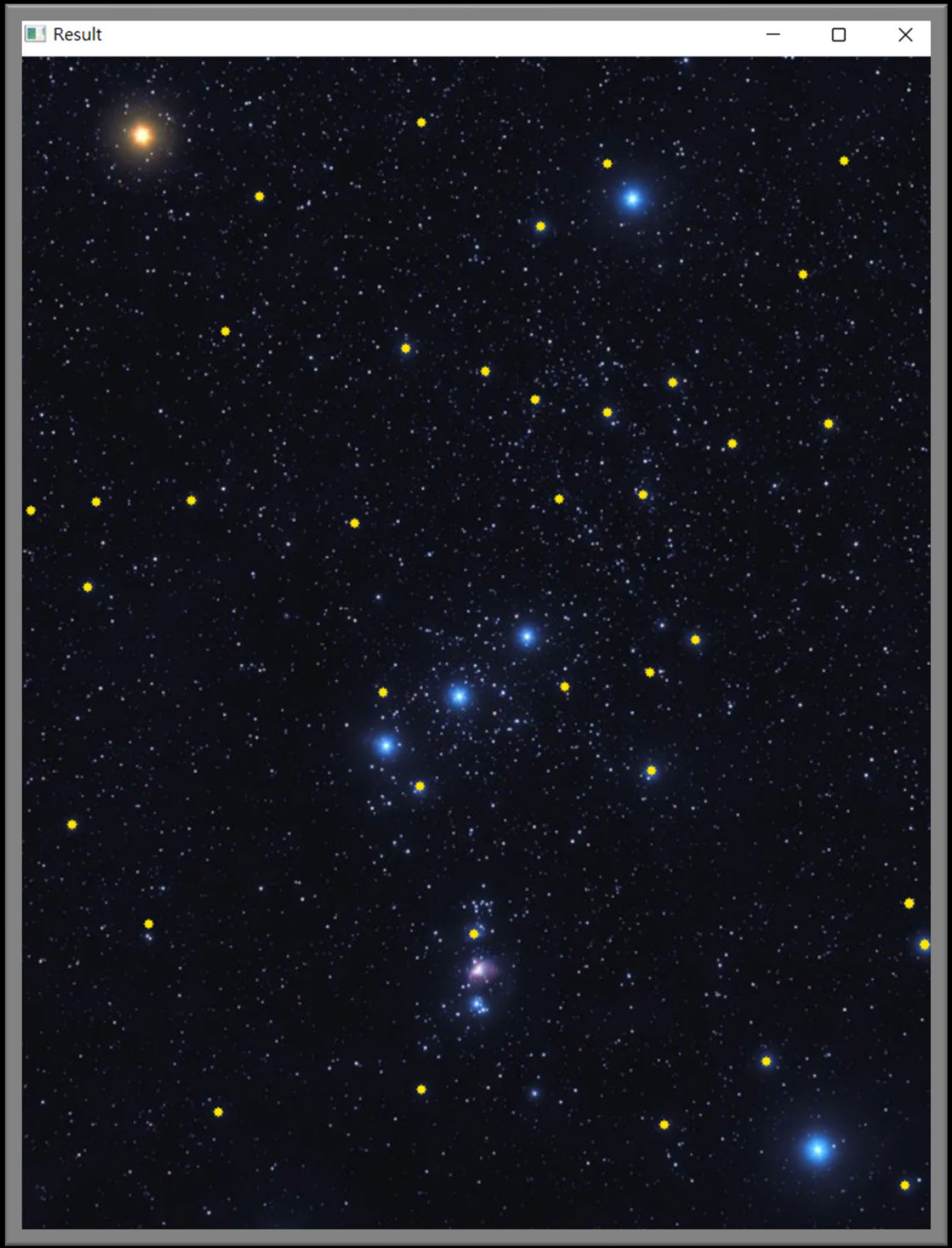


```python
if __name__ == "__main__":
    image = contourFinder(input("Input file name: "))
    t1 = float(input("Input Threshold 1: "))
    t2 = float(input("Input Threshold 2: "))
    image.findContours(t1, t2)

    cv.imshow("Result", image.clone)
    cv.waitKey(0)
```



```
Input file name: orion2.jpg
Input Threshold 1: 200
Input Threshold 2: 400
Sharpening?
```



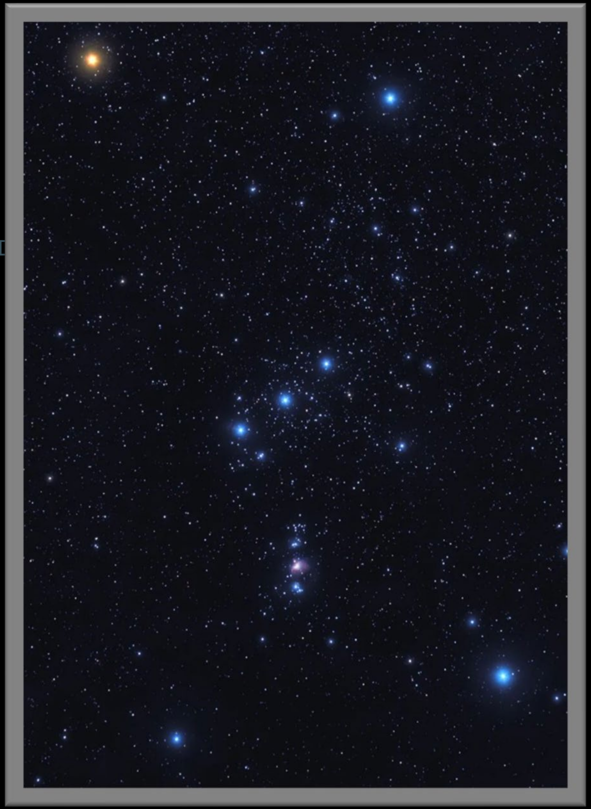


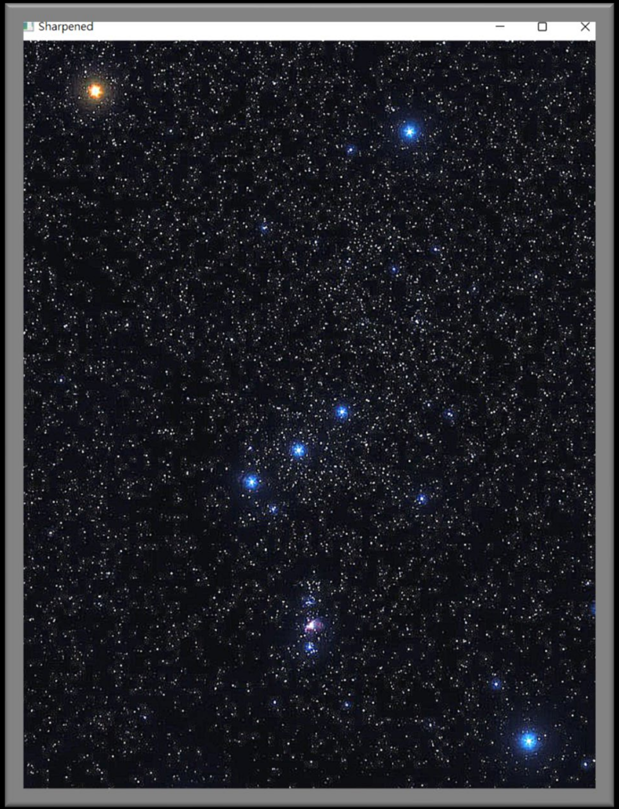
```
Input file name: orion3.jpg
Input Threshold 1: 200
Input Threshold 2: 400
Sharpening?
```

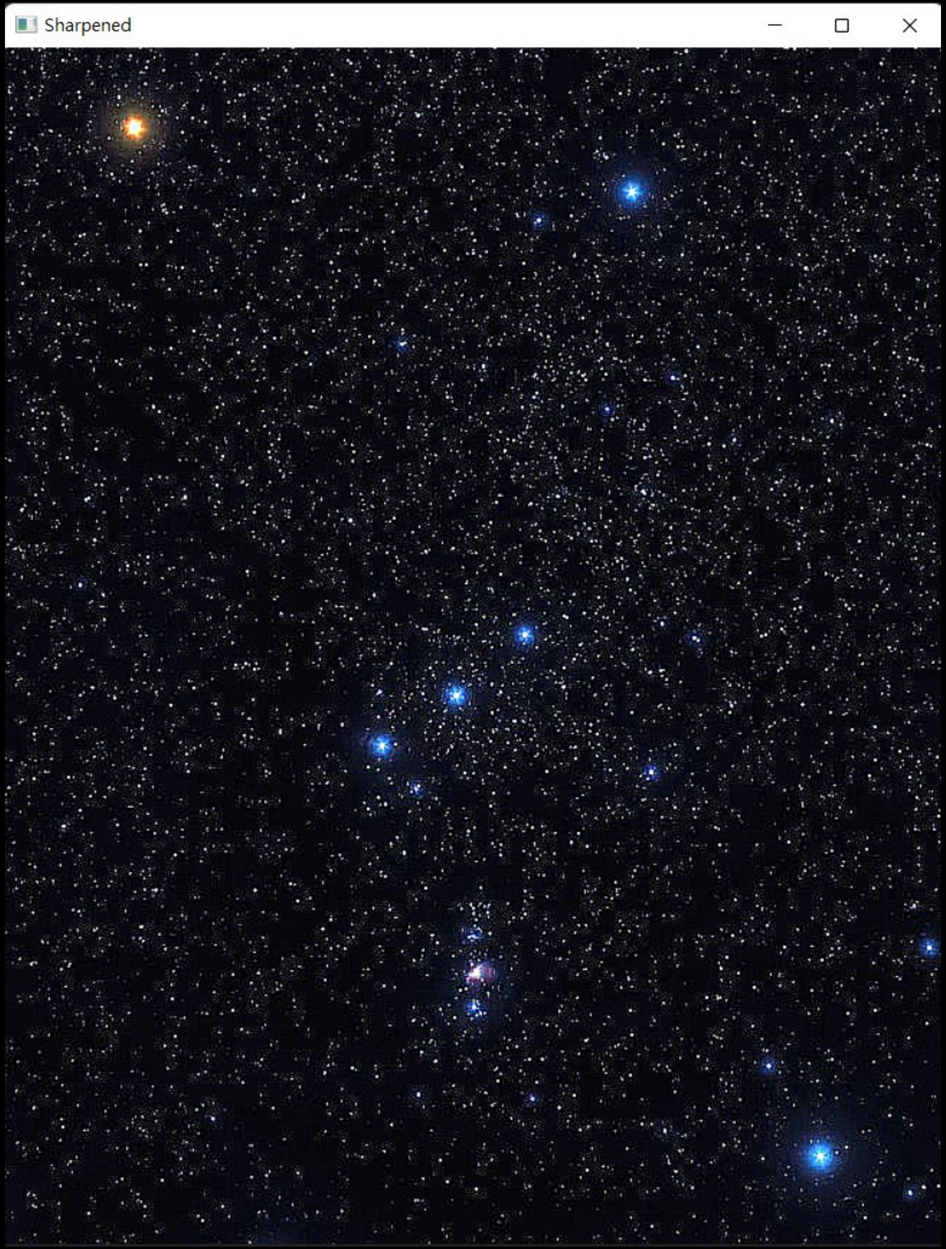
銳利化照片

```
class contourFinder:
    def __init__(self, filename) -> None:
        self.__image = cv.imread(str(filename))
        self.__preProcessing()

    def __preProcessing(self):
        self.__gray = cv.cvtColor(self.__image, cv.COLOR_BGR2GRAY)
        self.__blurred = cv.GaussianBlur(self.__gray, (5, 5), 1)

    def __sharpenImage(self):
        __kernel = np.array([[-1,-1,-1],
                             [-1, 9,-1],
                             [-1,-1,-1]])
        self.__sharpened = cv.filter2D(self.__image, -1, __kernel)

        return self.__sharpened

    def findContours(self, t1, t2):
        if input("Sharpening?")  == "Sharpened": tmp = self.__sharpenImage()
        else: tmp = self.__blurred

        self.__canneyed = cv.Canny(tmp, t1, t2)
        (self.cnts, _) = cv.findContours(self.__canneyed, cv.RETR_LIST, cv.CHAI
        self.clone = copy.deepcopy(self.__image)

        for c in self.cnts:
            M = cv.moments(c)

            cX, cY = (int(M["m10"]/M["m00"]), int(M["m01"]/M["m00"]))

            cv.circle(self.clone, (cX, cY), 3, (1, 227, 254), -1)
```

銳利化矩陣





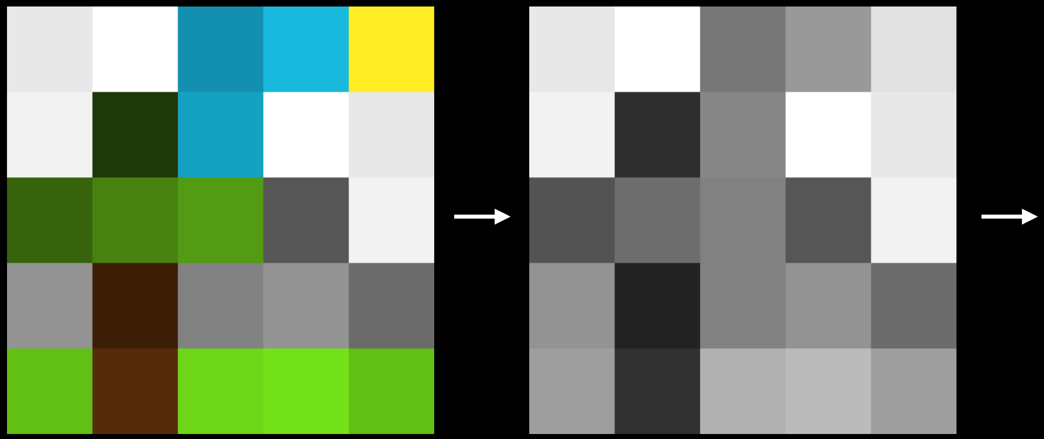

Sharpened

```
#creating custom dataset from directory
h, w = 1080, 1920

ds_train = tf.keras.preprocessing.image_dataset_from_directory(
    "/content/drive/MyDrive/Train",
    labels = "inferred",
    label_mode = "int",
    color_mode = "grayscale",
    batch_size = 20,
    image_size = (h, w),
    shuffle = True,
    seed = 123,
    validation_split = 0.1,
    subset = "training"
)
ds_validation = tf.keras.preprocessing.image_dataset_from_directory(
    "/content/drive/MyDrive/Train",
    labels = "inferred",
    label_mode = "int",
    color_mode = "grayscale",
    batch_size = 20,
    image_size = (h, w),
    shuffle = True,
    seed = 123,
    validation_split = 0.1,
    subset = "validation"
)
```

把照片轉成資料庫

```
#preprocessing and classifier
IMG_SIZE = 600 # All images will be resized to 600x600

def format (image, label):
 # returns an image that is reshaped to IMG_SIZE

  image = tf.cast(image, tf.float32)
  image = (image/127.5) - 1
  image = tf.image.resize(image, (IMG_SIZE, IMG_SIZE))
  return image, label

ds_train = ds_train.map(format)
ds_validation = ds_validation.map(format)

#calssifier
classname = ["Aquarius", "Aries", "Cancer", "Capricorn", "Gemini", "Leo", "Libra", "Pisces", "Sagittarius", "Scorpio", "Taurus", "Virgo"]
```

降畫質，縮小數值



200	255	80	120	180
220	10	110	255	200
80	95	110	80	250
120	10	110	120	110
120	50	130	150	120

0.568627	1.00000	-0.37255	-0.05882	0.411765
0.72549	-0.92157	-0.13725	1.00000	0.568627
-0.37255	-0.2549	-0.13725	-0.37255	0.960784
-0.05882	-0.92157	-0.13725	-0.05882	-0.13725
-0.05882	-0.60784	0.019608	0.176471	-0.05882

我們把每張星星的圖轉成灰階之後，轉換成電腦可以理解的數值陣列，並且把數值縮小到-1~1的區間。

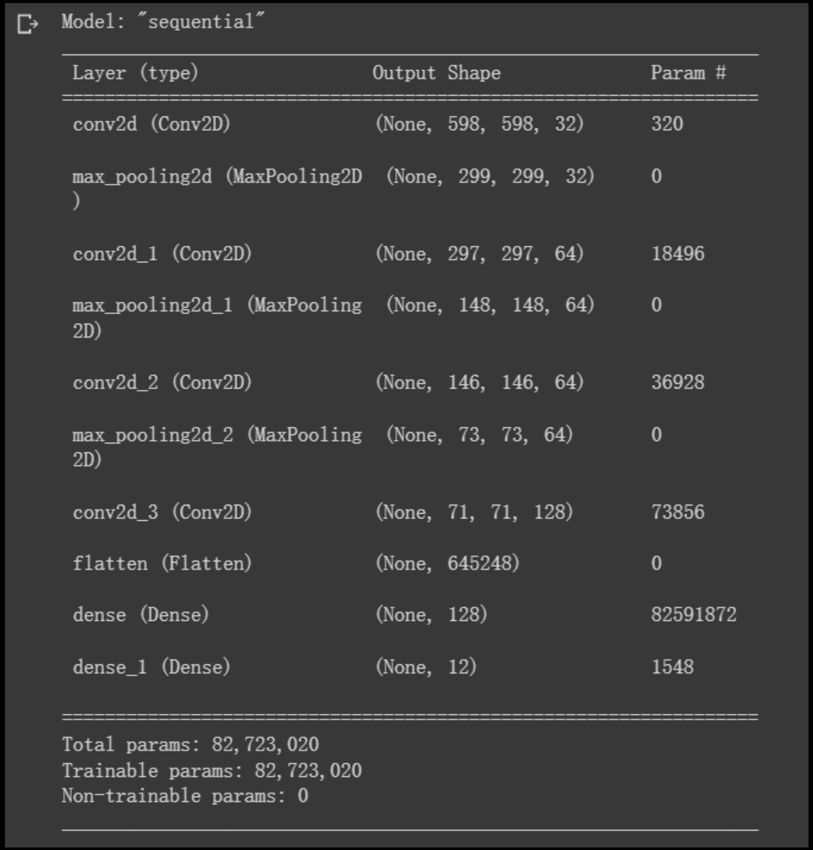
```
#convolutional layers
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(600, 600, 1)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
```

卷積層

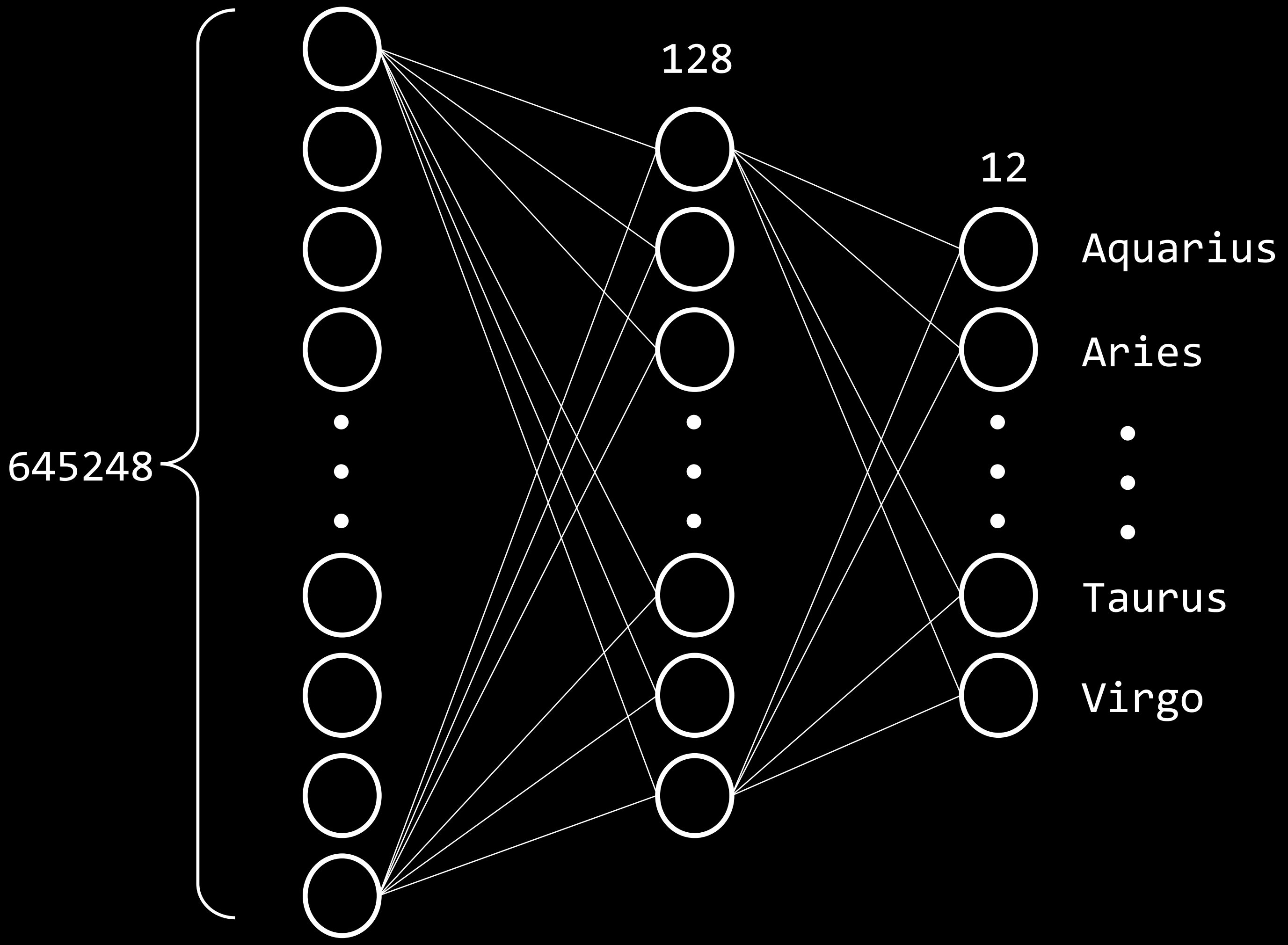
```
#dense neural network
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(12))
```

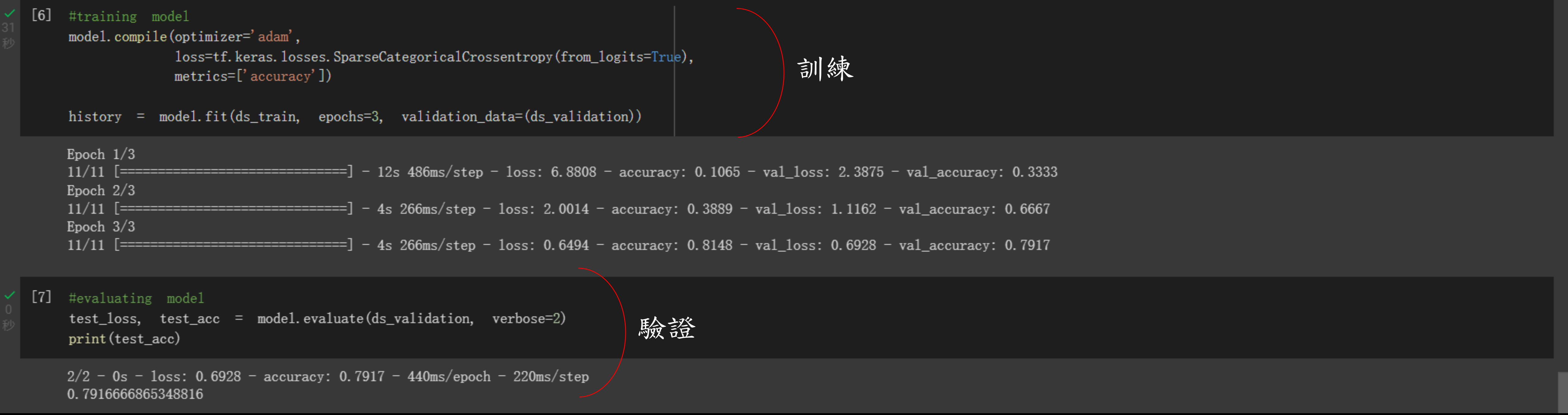
神經網路層

建構卷積神經網路，四層卷積層可以提取出照片的特徵，最後神經網路輸出的12個神經元就是相對應的星座。



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 598, 598, 32)      320

 max_pooling2d (MaxPooling2D  (None, 299, 299, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 297, 297, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 148, 148, 64)     0
 2D)

 conv2d_2 (Conv2D)           (None, 146, 146, 64)      36928

 max_pooling2d_2 (MaxPooling  (None, 73, 73, 64)       0
 2D)

 conv2d_3 (Conv2D)           (None, 71, 71, 128)       73856

 flatten (Flatten)           (None, 645248)            0

 dense (Dense)               (None, 128)               82591872

 dense_1 (Dense)             (None, 12)                1548

=================================================================
Total params: 82,723,020
Trainable params: 82,723,020
Non-trainable params: 0
_________________________________________________________________
```





```
[6] #training  model
    model.compile(optimizer='adam',
                  loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
                  metrics=['accuracy'])

    history  =  model.fit(ds_train,   epochs=3,   validation_data=(ds_validation))
```

```
Epoch 1/3
11/11 [==============================] - 12s 486ms/step - loss: 6.8808 - accuracy: 0.1065 - val_loss: 2.3875 - val_accuracy: 0.3333
Epoch 2/3
11/11 [==============================] - 4s 266ms/step - loss: 2.0014 - accuracy: 0.3889 - val_loss: 1.1162 - val_accuracy: 0.6667
Epoch 3/3
11/11 [==============================] - 4s 266ms/step - loss: 0.6494 - accuracy: 0.8148 - val_loss: 0.6928 - val_accuracy: 0.7917
```

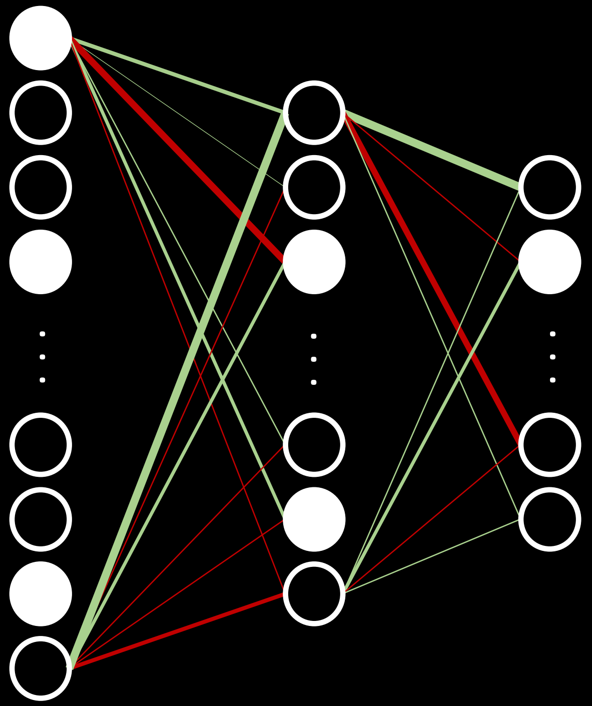
訓練

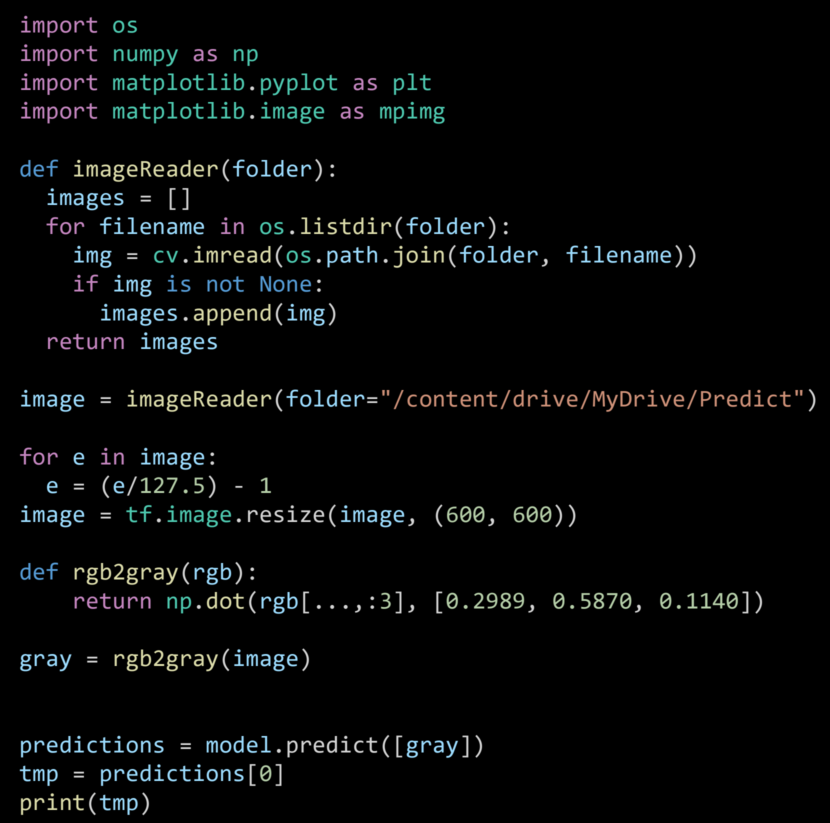
```
[7] #evaluating  model
    test_loss,  test_acc  =  model.evaluate(ds_validation,   verbose=2)
    print(test_acc)
```

```
2/2 - 0s - loss: 0.6928 - accuracy: 0.7917 - 440ms/epoch - 220ms/step
0.7916666865348816
```

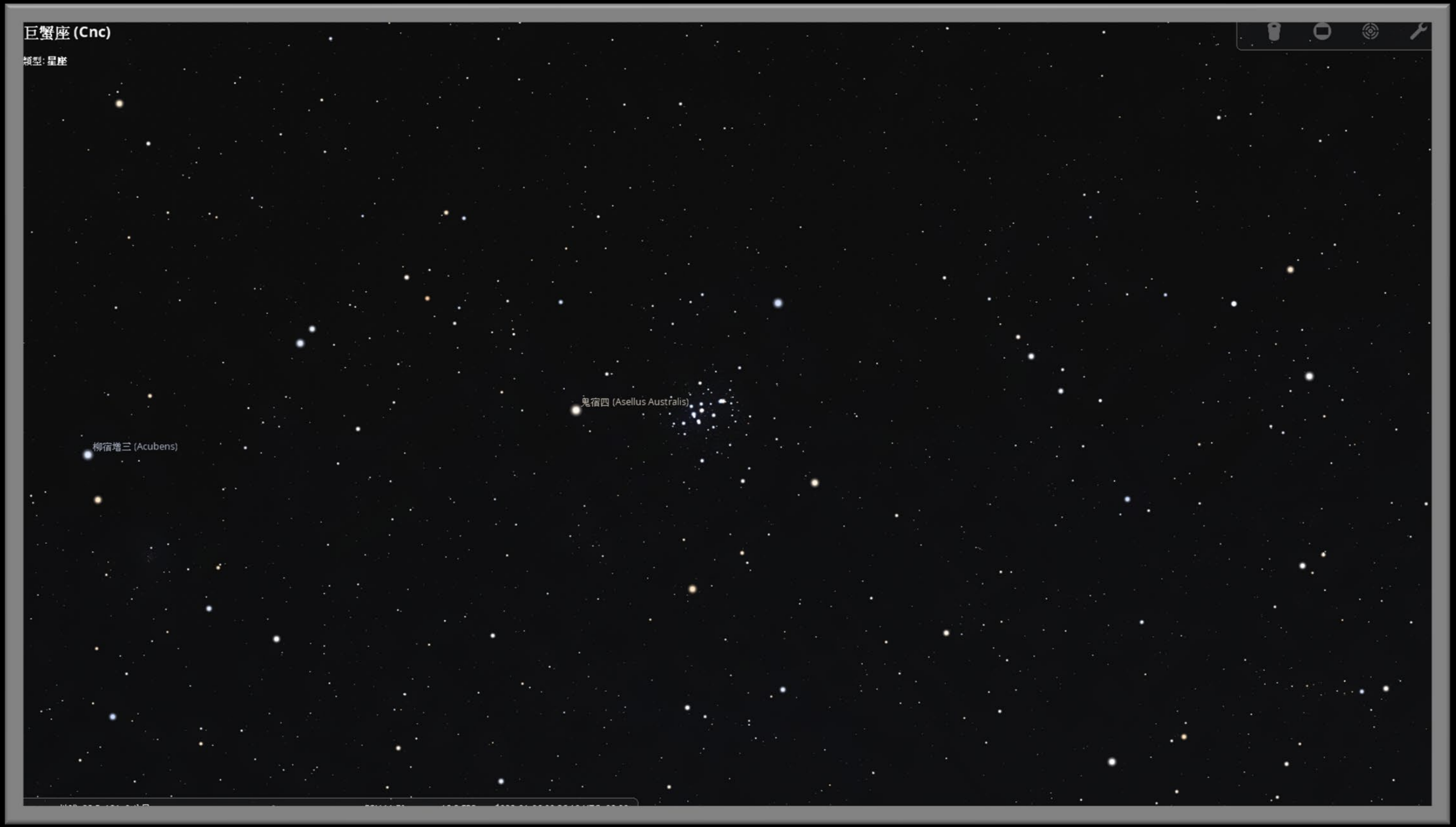
驗證

透過剛建置好的神經網路，讓電腦學習每張照片的特徵，調整神經連結的權重，訓練他在未來收到一張照片的時候能夠辨識出是哪個星座。

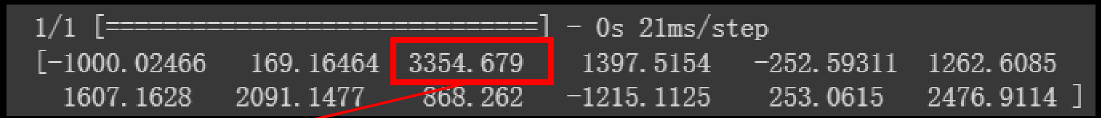




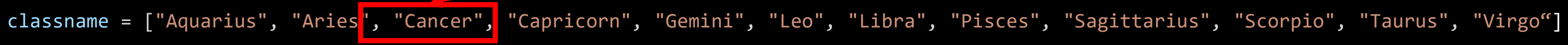
```
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

def imageReader(folder):
  images = []
  for filename in os.listdir(folder):
    img = cv.imread(os.path.join(folder, filename))
    if img is not None:
      images.append(img)
  return images

image = imageReader(folder="/content/drive/MyDrive/Predict")

for e in image:
  e = (e/127.5) - 1
image = tf.image.resize(image, (600, 600))

def rgb2gray(rgb):
    return np.dot(rgb[...,:3], [0.2989, 0.5870, 0.1140])

gray = rgb2gray(image)


predictions = model.predict([gray])
tmp = predictions[0]
print(tmp)
```



真的把一張照片輸入神經網路，修改成電腦能讀取的格式後，匯進訓練完成的神經網路，發現輸出的結果是正確的，成功辨識出星座種類。



```
1/1 [==============================] - 0s 21ms/step
[-1000.02466    169.16464   3354.679     1397.5154    -252.59311   1262.6085
  1607.1628    2091.1477     868.262    -1215.1125     253.0615    2476.9114 ]
```



```
classname = ["Aquarius", "Aries", "Cancer", "Capricorn", "Gemini", "Leo", "Libra", "Pisces", "Sagittarius", "Scorpio", "Taurus", "Virgo“]
```

問題與討論

這次的專案遇到了許多問題，其中有些問題仍然沒有辦法解決:

1. 我們在測試時，發現現實中的照片效果不佳，因為大氣層的緣故，星星拍出來會有嚴重的光暈，造成在辨識亮星的時候會有困難。如簡報裡描述，我們曾經試過用銳利化矩陣嘗試解決這個問題，無奈效果不佳，也暫時沒有想出新的解決方法。

2. 由於我們用來訓練的照片需求太特殊，網路上幾乎沒有任何建構好的資料庫，所以我們每張照片都是徒手擷取，也導致我們照片在太少張。12個星座扣除認證用途的照片後，僅剩下216張訓練用的照片，造成訓練的效果不佳，信心度僅79%，而且也不能完整訓練天球88個星座。

3. 實作過程中，遇到了Google Colab硬體限制上的問題，常常因為記憶體用盡導致執行終止，為了降低計算量我們有調整每張照片的比例跟畫素，但這可能造繩一些資訊的流失，也希望未來可以解決這個問題。

結論與展望

我們成功以Tensorflow與OpenCV實踐從照片辨識星座，雖然因為照片的取得困難我們僅成功訓練辨識黃道12宮的星座，但未來如果有辦法取得大量照片訓練，相信有辦法改善信心度，且我們的想法有辦法推廣到全天的星座。

未來希望可以研發出一款APP，在導入模型後可以利用相機直接判斷星座，為使用者導覽星空

我們參考的文獻指出，這種視覺的星座辨識可以應用於小型衛星的定位，他提出可以利用視覺系統辨識特定星星圖樣，有助於減輕衛星重量。

專案的程式碼：https://github.com/CYHuang0429/TFstar.git

參考資料

- Daniel Hingston, 2019. *Development of a Computer Vision Based Orientation System for CubeSats*, University of Strathclyde
  Github Repo: https://github.com/raspberrystars/CV-Star-Sensor.git
- TensorFlow 2.0 Complete Course - Python Neural Networks for Beginners Tutorial: https://www.youtube.com/watch?v=tPYj3fFJGjk
- Stack Overflow
- OpenCV documentation: https://docs.opencv.org/3.4/