

Chapter 4 : TM X Confusion Matrix

賴秉樑 debugger

學院創辦人

課程網址 <https://max543.com/debugger>

Outline

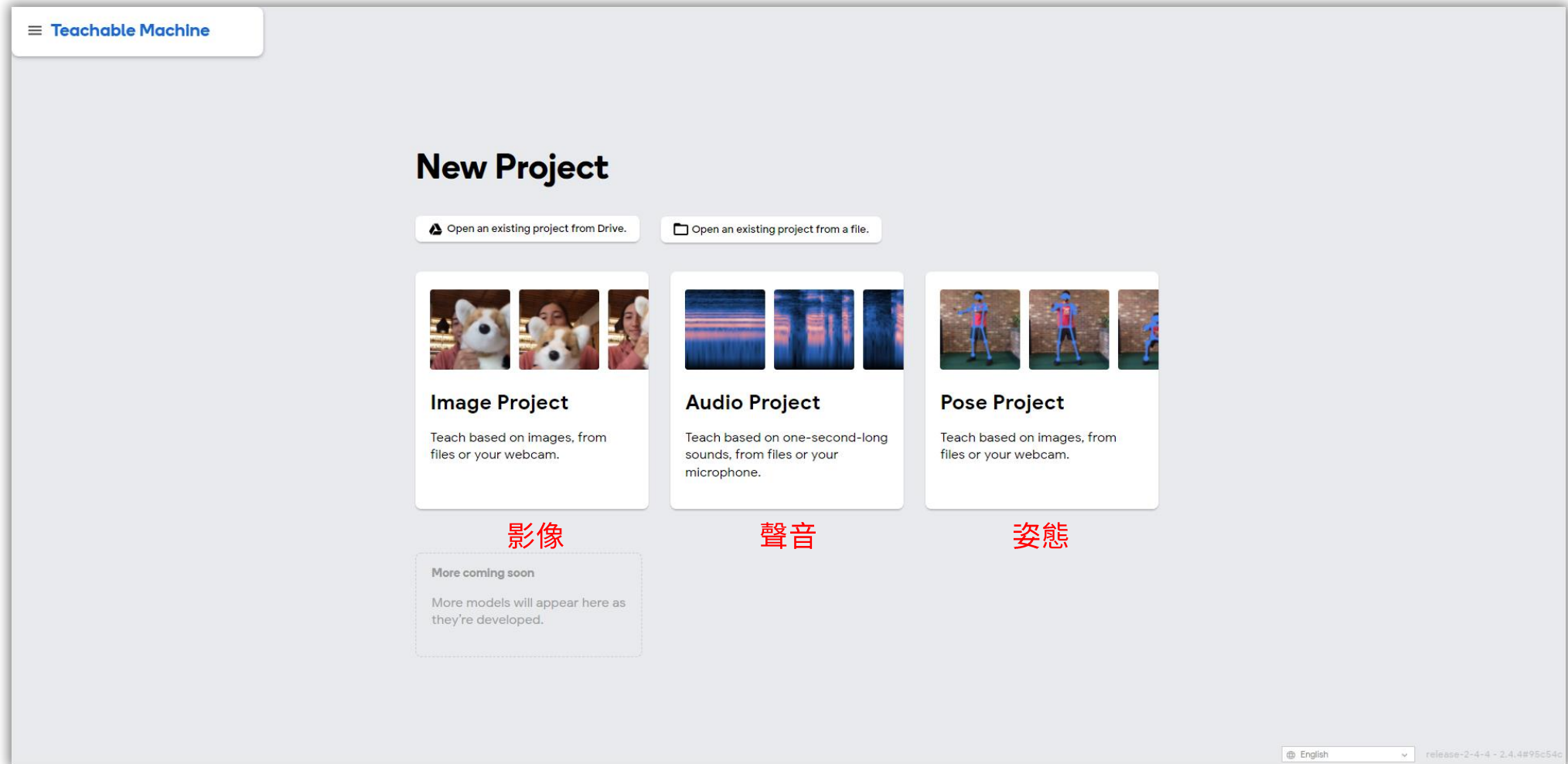
- TM 平臺訓練模型
- 下載模型
- 在 Python 中應用模型，使用 Webcam 即時分析

Teachable Machine

- Teachable Machine (以下簡稱 TM) 是 Google 推出的平台，Google 的強項是肢體辨識，但 Google 目前只能做分類器。
- 採用 Tensorflow 的 keras 的神經網路模型。

Google Teachable Machine

- 官網：<https://teachablemachine.withgoogle.com/>



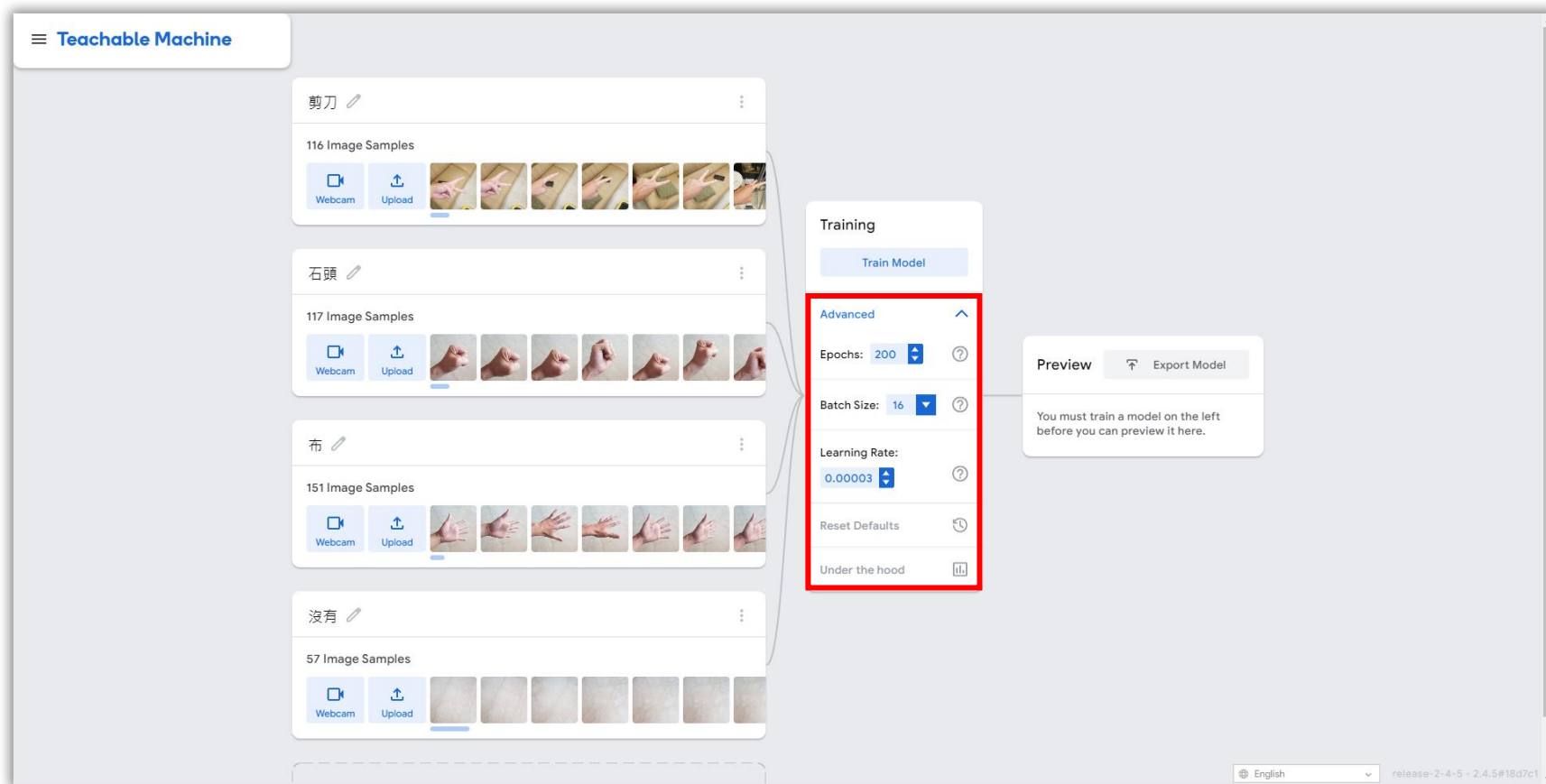
Step 1：加入資料

- 在此頁面中可以看到整個流程只有三個步驟：1. 資料 2. 訓練 3. 預測。

The screenshot displays the Teachable Machine interface. On the left, there are four data categories: '剪刀' (Scissors) with 116 Image Samples, '石頭' (Stone) with 117 Image Samples, '布' (Cloth) with 151 Image Samples, and '沒有' (None) with 57 Image Samples. Each category has a 'Webcam' and 'Upload' button, and a row of image thumbnails. A red box highlights the '剪刀' category. On the right, the 'Training' panel is active, showing a 'Train Model' button and a dropdown menu set to 'Advanced'. The 'Preview' panel is inactive, displaying a message: 'You must train a model on the left before you can preview it here.' The interface also includes a language dropdown set to 'English' and a version number 'release-2-4-5 - 2.4.5#18d7c1'.

Step 2 : 訓練

- 點選 Advance 進階，可以看到有 Epochs (世代) : 50、Batch Size (批次) : 16、Learning Rate (學習率) : 0.001，一般來說，如果不是太複雜的東西，這三個選項可以不用調整。



Step 3：預測

- 完成後，會開啟預覽視窗，讓你測試辨識效果。

The screenshot displays the Teachable Machine interface during the prediction phase. On the left, there are four training categories: 剪刀 (Scissors), 石頭 (Rock), 布 (Paper), and 沒有 (None). Each category has a 'Webcam' and 'Upload' button, and a row of 15 image samples. The 'Training' panel in the center shows 'Model Trained' and advanced settings: Epochs: 50, Batch Size: 16, and Learning Rate: 0.001. On the right, the 'Preview' window is highlighted with a red border, showing a live webcam feed of a hand in a scissors gesture. Below the feed, the 'Output' section shows a progress bar for '剪刀' at 98%, and bars for '石頭', '布', and '沒有' are empty. The interface also includes a language dropdown set to 'English' and a version number 'release-2-4-5 - 2.4.5#18d7c1'.

分析結果

- 如果想看更細部的模型學習過程，可以點選 Under the hood 查看，這是混淆矩陣，可以查看錯誤分佈。

The screenshot displays the Teachable Machine interface. On the left, there are four categories: 剪刀 (Scissors), 石頭 (Rock), 布 (Paper), and 沒有 (None), each with 15 image samples. The central 'Training' panel shows 'Model Trained' and 'Advanced' settings: Epochs: 50, Batch Size: 16, and Learning Rate: 0.001. A red box highlights the 'Under the hood' icon in the training panel, with an arrow pointing to the analysis results on the right.

The analysis results on the right include:

- Accuracy per class:**

CLASS	ACCURACY	# SAMPLES
剪刀	1.00	3
石頭	1.00	3
布	1.00	3
沒有	1.00	3

- Confusion Matrix:**

Class	剪刀	石頭	布	沒有
剪刀	3	0	0	0
石頭	0	3	0	0
布	0	0	3	0
沒有	0	0	0	3

- Accuracy per epoch:**

混淆矩陣 (Confusion Matrix)

用來評估模型好壞常見的方法

一般型的 Confusion Matrix (1/2)

True/False
正確與否?

Positive/Negative
預測方向

	實際 True	實際 False
預測 True	TP (True Positive)	FP (False Positive) Type I Error
預測 False	FN (False Negative) Type II Error	TN (True Negative)

- TP (正確的正向預測)：實際有病，醫生預測有病。(越多越好)
- TN (正確的負向預測)：實際無病，醫生預測無病。(越多越好)
- FN (錯誤的負向預測)：實際有病，醫生預測無病。(越少越好) (大門門禁若有錯誤，希望是這種。廣告投放卻不想)
- FP (錯誤的正向預測)：實際無病，醫生預測有病。(越少越好) (大門門禁若有錯誤，不希望是這種。廣告投放要這種)

一般型的 Confusion Matrix (2/2)

	True/False 正確與否?	Positive/Negative 預測方向
	實際 True	實際 False
預測 True	TP (True Positive)	FP (False Positive) Type I Error
預測 False	FN (False Negative) Type II Error	TN (True Negative)

- Accuracy (準確度) : $(TP + TN) / \text{所有情形的個數}$ ，這是最常用的指標。
- Precision (精密度) : $TP / (TP + FP)$ (預測為正向時，實際的的準確度有多少?)
 - ✓ 例：大門門禁，在意的是預測正向（開門）的答對多少，比較不在意實際正向（是主人）的答對多少。
- Recall (召回率) : $TP / (TP + FN)$ (實際為正向時，實際的的準確度有多少?)
 - ✓ 例：廣告投放，在意的是實際正向（是潛在客戶）的答對多少，而相對比較不在意預測正向（廣告投出）答對多少。

TM 用的名詞解釋

- 以猜拳為例：(假設剪刀、石頭、布、沒有，各有 15 張照片)
 - ✓ Training samples : 85% of samples (訓練樣本，使用 12 張)
 - ✓ Test samples : 15% of samples (測試樣本，使用 3 張)
 - ✓ Accuracy (又稱之為 Recall 召回率) : $TP / \text{Samples per Class}$ (TM 所採用的定義)

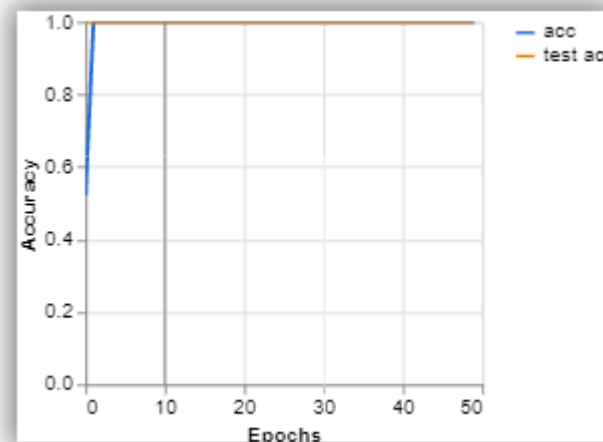
TM - 四種分析結果

$$\text{Accuracy} = \text{TP} / \text{Samples per Class}$$

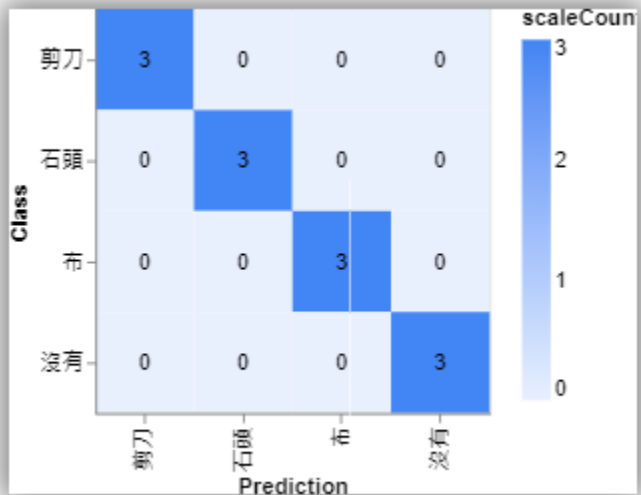
Accuracy per calss

CLASS	ACCURACY	# SAMPLES
剪刀	1.00	3
石頭	1.00	3
布	1.00	3
沒有	1.00	3

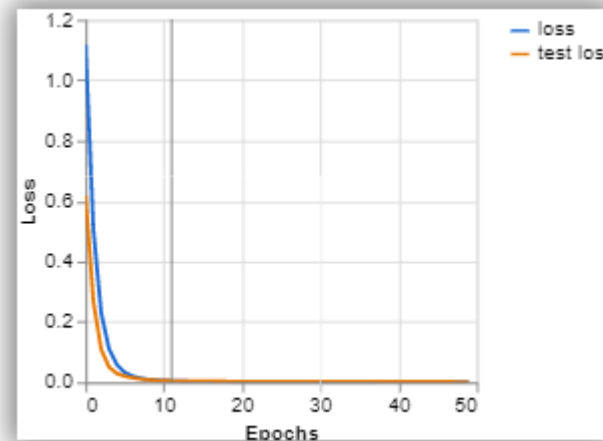
Accuracy per epoch



Confusion Matrix



Loss per epoch



Confusion Matrix of TM

The screenshot displays the Teachable Machine interface. On the left, there are four panels for training classes: 剪刀 (Scissors), 石頭 (Rock), 布 (Paper), and 沒有 (Nothing). Each panel shows 15 image samples. A red box highlights a set of hand images under the 'Paper' class, with the text '混淆的訓練樣本' (Confused training samples) written in red above them. The central 'Training' panel shows 'Model Trained' and advanced settings: Epochs: 50, Batch Size: 16, Learning Rate: 0.001. The 'Preview' window shows a dark image. The 'Output' section shows accuracy for each class: 剪刀 (33%), 石頭 (63%), 布 (0%), and 沒有 (63%). On the right, the 'Accuracy per class' table and 'Confusion Matrix' are shown. The table has a red box around the 'Paper' row, and the confusion matrix has a red box around the 'Paper' row.

CLASS	ACCURACY	# SAMPLES
剪刀	1.00	3
石頭	1.00	3
布	0.67	3
沒有	1.00	3

Class	剪刀	石頭	布	沒有
剪刀	3	0	0	0
石頭	0	3	0	0
布	1	0	2	0
沒有	0	0	0	3

Confusion Matrix：檢查模型與測試的詳細結果，以修正 Accuracy。

Conclusion

- 由一般型的 Confusion Matrix，衍生許多的應用參數。
 - ✓ 可視情況採用或修正參數。
 - ✓ 訓練模型時，可供參考與除錯。
- 參考資料
 - ✓ <https://www.ycc.idv.tw/confusion-matrix.html>

Outline

- TM 平臺訓練模型
- 下載模型
- 在 Python 中應用模型，使用 Webcam 即時分析

下載模型

The image shows a screenshot of the Teachable Machine web interface. In the top right, the 'Export Model' button is highlighted with a red box. A modal dialog box is open in the foreground, titled 'Export your model to use it in projects.' It has three tabs: 'Tensorflow.js', 'Tensorflow', and 'Tensorflow Lite'. Under the 'Tensorflow' tab, there are two radio buttons for 'Model conversion type': 'Keras' (which is selected and also highlighted with a red box) and 'Savedmodel'. A 'Download my model' button is also highlighted with a red box. Below the radio buttons, there is a text description: 'Converts your model to a keras .h5 model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.' Underneath, there is a section for 'Code snippets to use your model:' with a 'Keras' sub-section and a 'Contribute on Github' link. A code editor contains the following Python code:

```
from keras.models import load_model
from PIL import Image, ImageOps
import numpy as np

# Load the model
model = load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
# Replace this with the path to your image
image = Image.open('<IMAGE_PATH>')
#resize the image to a 224x224 with the same strategy as in TM2:
#resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
```

The background interface shows a 'Model Trained' status, 'Advanced' settings with 'Epochs' set to 200, and a 'Preview' section with an 'Input' toggle set to 'ON' and a 'Webcam' dropdown menu. On the left, there are several image sample galleries with titles like '剪刀', '石頭', '布', and '沒有', each with 'Webcam' and 'Upload' buttons.

Outline

- TM 平臺訓練模型
- 下載模型
- 在 Python 中應用模型，使用 Webcam 即時分析

所需 Python 的模組

- opencv
 - ✓ 請參考前面上課講義。
- tensorflow
 - ✓ `pip install tensorflow`

整合專案：使用 Webcam 即時分析

Demo

converted_keras

請參考資料夾內的程式碼