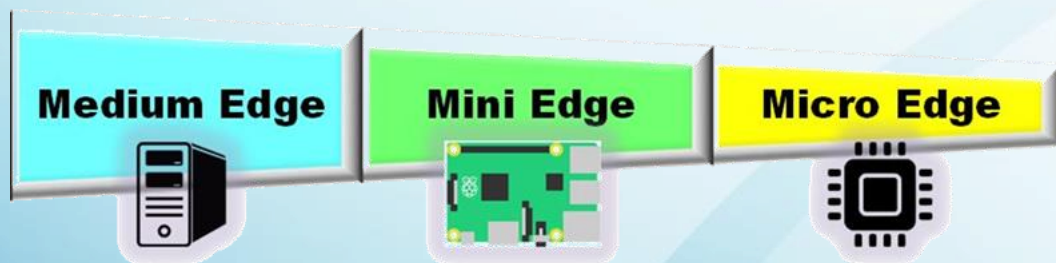
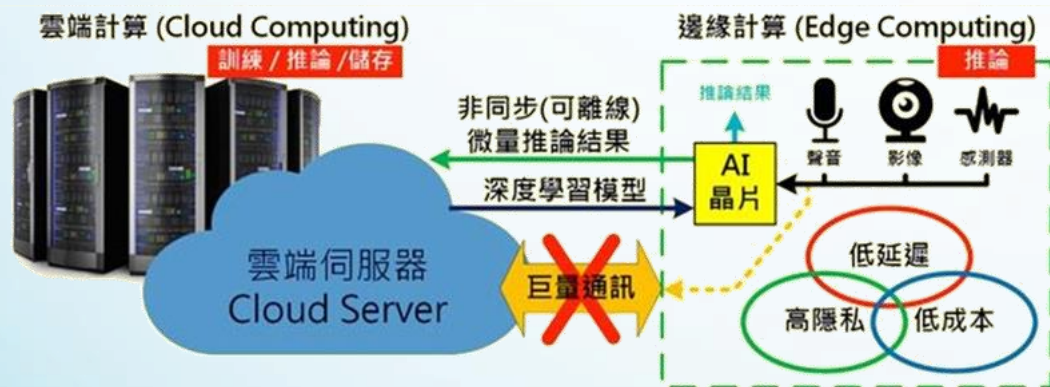


OmniXRI's Edge AI & TinyML 小學堂



歡迎加入
邊緣人俱樂部



【第4講】

開源模型訓練工具



歐尼克斯實境互動工作室 (OmniXRI Studio)
許哲豪 (Jack Hsu)

簡報大綱



- 4.1. AI 工作流程
- 4.2. 開源訓練工具

本課程完全免費，請勿移作商業用途！
歡迎留言、訂閱、點讚、轉發，讓更多需要的朋友也能一起學習。

完整課程大綱：<https://omnixri.blogspot.com/2024/02/omnixris-edge-ai-tinyml-0.html>
課程直播清單：<https://www.youtube.com/@omnixri1784/streams>

4.1. AI 工作流程



- 問題定義
- 建立模型
- 訓練調參
- 佈署推論

問題定義 — 輸出入

輸入資料

非時序型資料

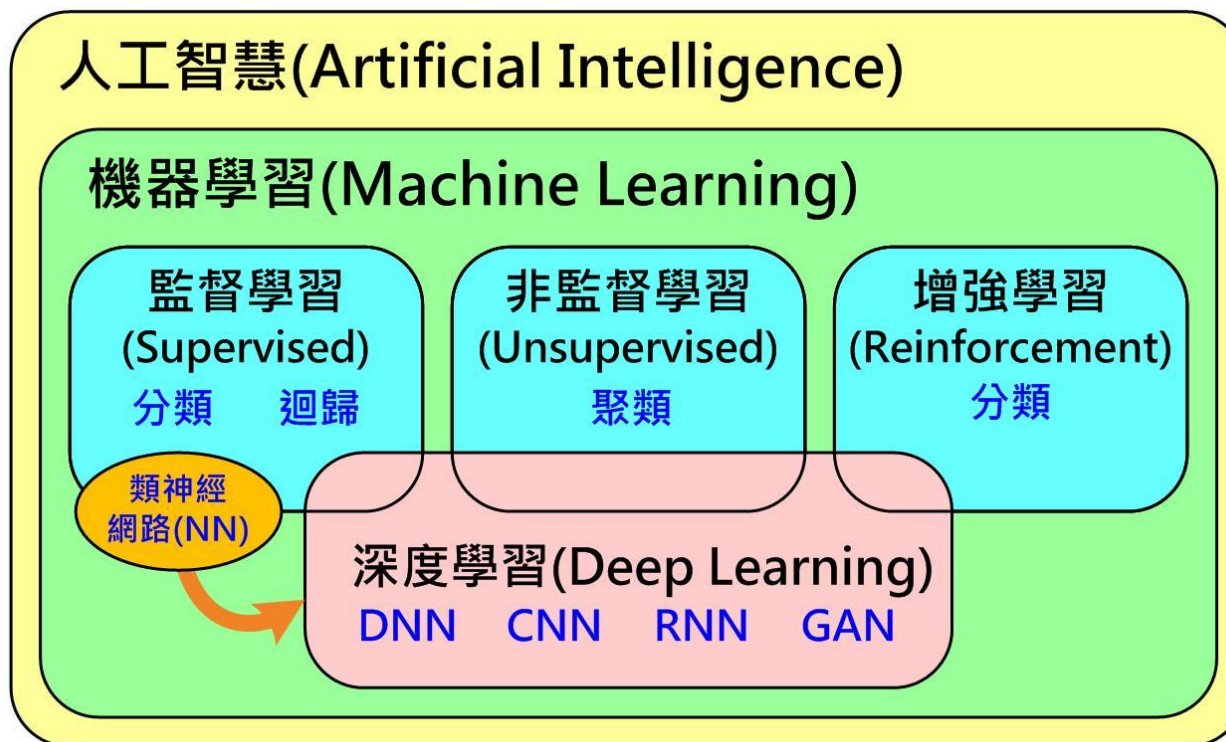
影像

時序型資料

視頻/語音

文章/數值

混合型資料



輸出結果

非時序型結果

分類/定位

時序型結果

動作/語意

摘要/預警

資料來源：<http://omnixri.blogspot.com/2018/05/aimaker.html>

問題定義 — 流程規畫

人工智慧（大數據）vs. 咖哩雞飯



問題定義—工作流程

確認資料集建構方向

資料收集、清洗、擴增、平衡

確認資料標註格式

標註規範、標註工具、
輸出格式、協同工作 ...

選用合適框架及模型

TensorFlow, PyTorch ...
分類、物件、分割、語言...

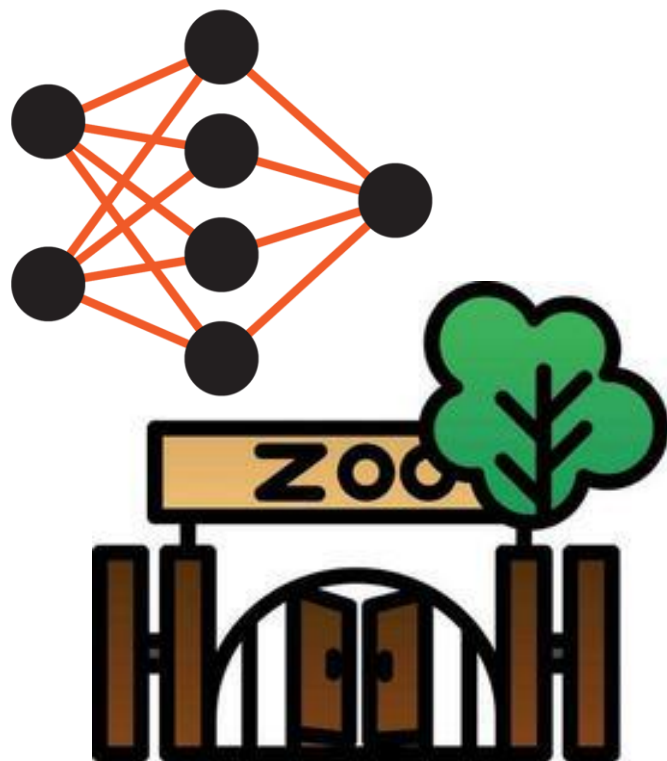
進行模型訓練與調參

超參數設定、自動調整、
效能評估、

模型優化及部署

量化、剪枝、蒸餾 ...
硬體選用、推論環境 ...

建立模型 — 模型動物園



- Model Zoo

<https://modelzoo.co/>

- OpenVINO Open Model Zoo

https://github.com/openvinotoolkit/open_model_zoo

- Kaggle Models

<https://www.kaggle.com/models>

- Papers with code Methods

<https://paperswithcode.com/methods>

- Roboflow Models

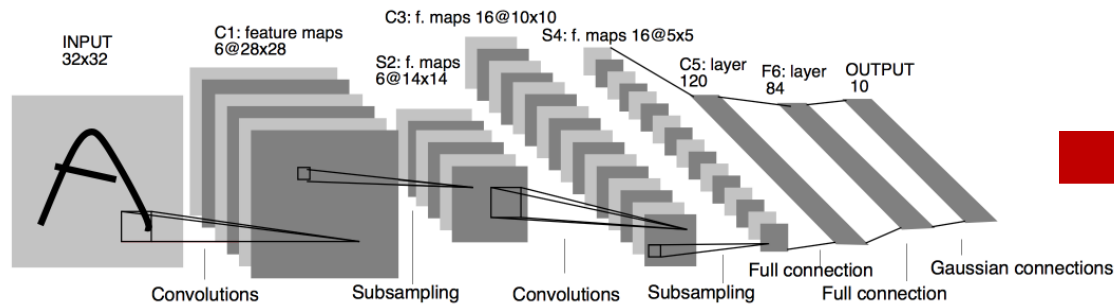
<https://roboflow.com/models>

- HuggingFace Models

<https://huggingface.co/models>

建立模型 — 自建模型

LeNet-5網路架構概念



選用開發框架



網路結構瀏覽器

NETRON

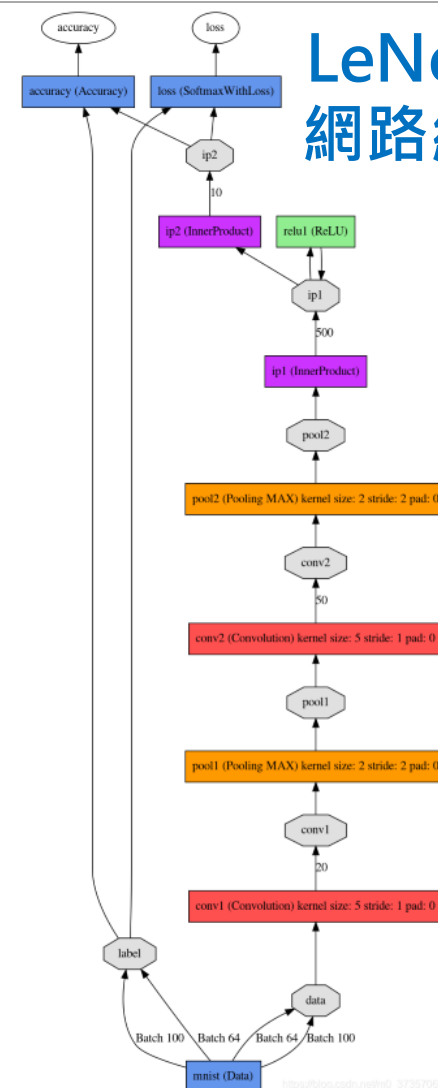
網頁版：

<https://netron.app/>

下載版：

<https://github.com/lutzroeder/netron>

LeNet-5 網路結構

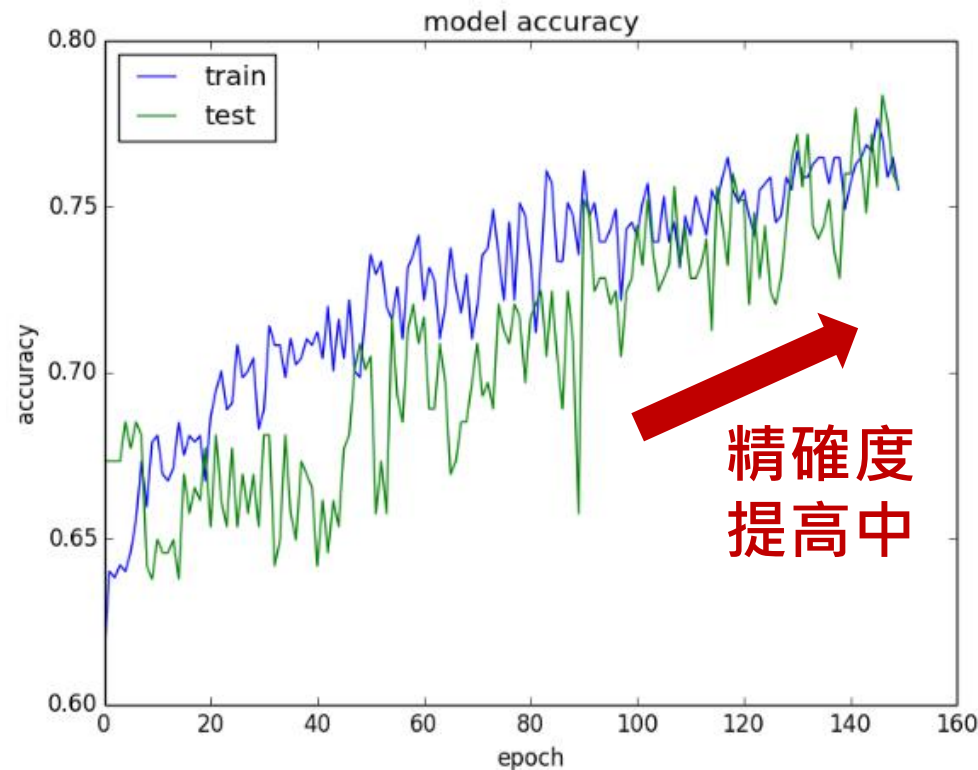
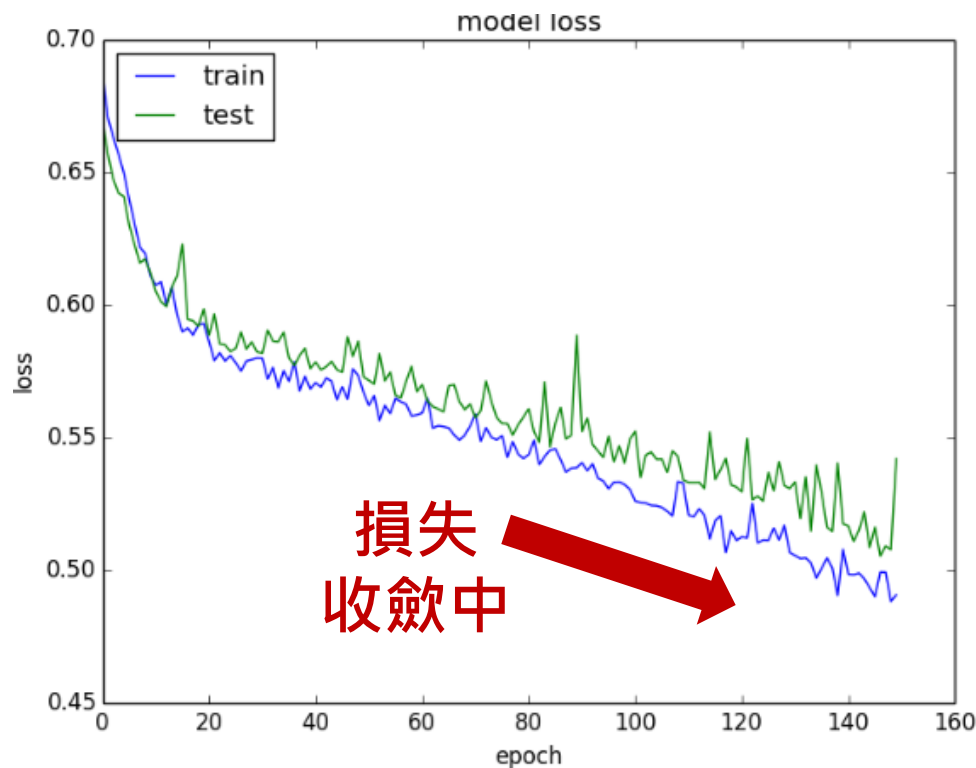


訓練調參－可視化工具

迭代損失圖

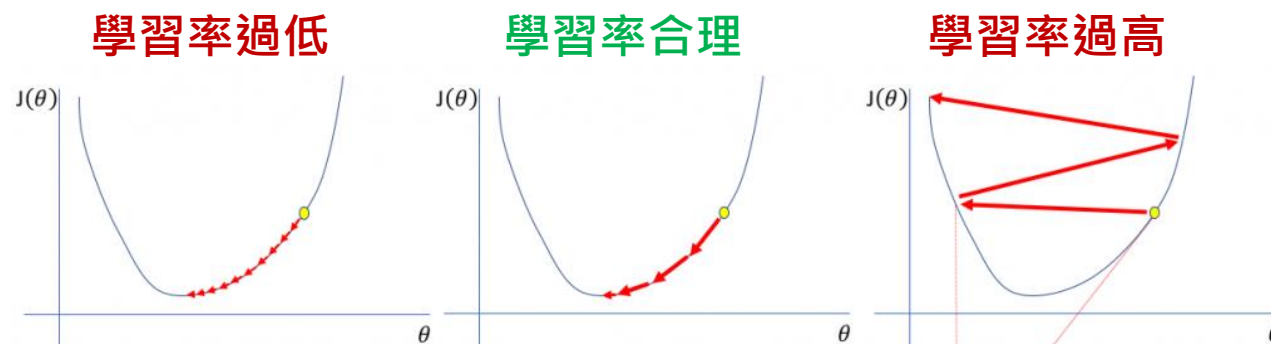
【監督式學習】

迭代精確度圖

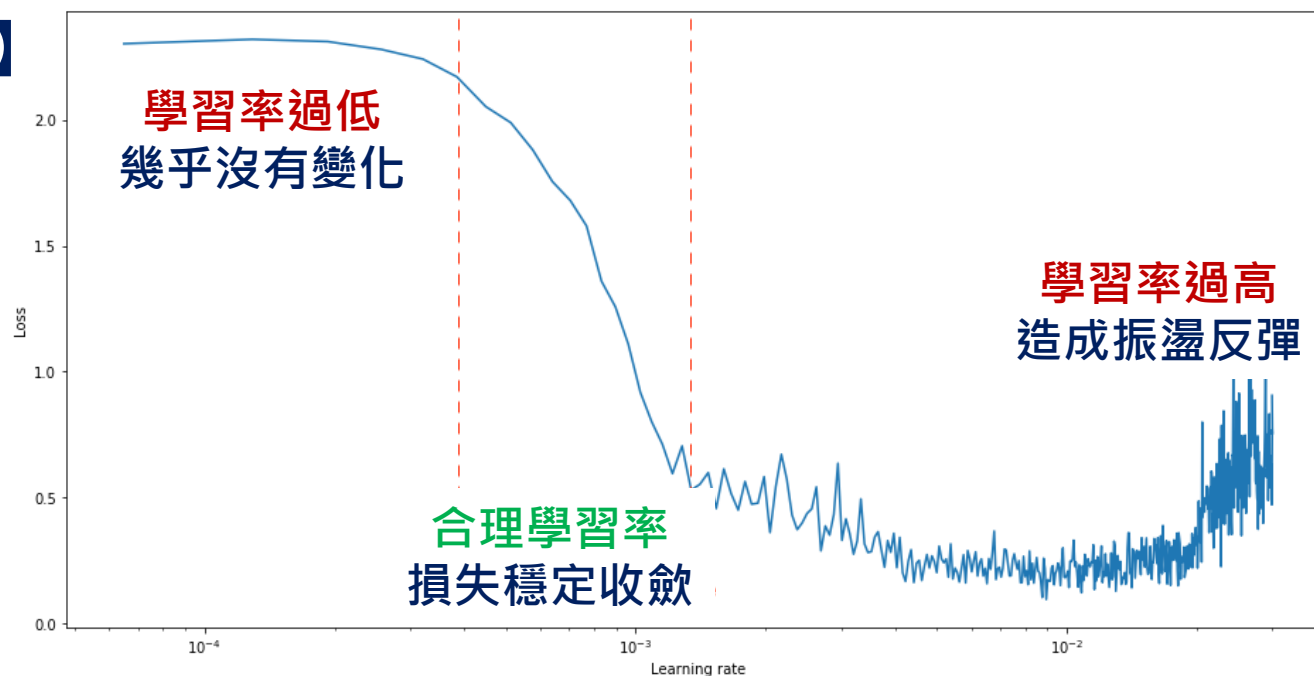


資料來源：<https://cnbeining.github.io/deep-learning-with-python-cn/4-advanced-multi-layer-perceptrons-and-keras/ch15-understand-model-behavior-during-training-by-plotting-history.html>

訓練調參 — 學習率



【監督式學習】



資料來源：<https://ithelp.ithome.com.tw/m/articles/10204032>

訓練調參 — 常見超參數

- 須人為介入調整的參數稱為「**超參數**」。
- **batch_size**：一次迭代放入進行訓練或測試的影像數量。
- **epoch**：所有資料皆被計算過1次後即為1 epoch。
- **learning Rate**：學習速率，可選自動調整類型。
- **base_lr**：初始學習率
- **step_size**：多少次迭代後下降學習率
- **gamma**：下降學習率的幅度
- **momentum**：動量
- **weight_decay**：權重的懲罰係數
- **max_iter**：訓練時的迭代次數

訓練調參 — 成果指標

客觀指標

- **分類問題**：正確率、精確率、召回率、F1分數、全類平均精確率(mAP)、ROC / PR曲線...
- **物件偵測**：IoU, mAP
- **影像分割**：PA, MPA, MIoU
- **回歸預測**：MSE, RMSE, MAE

主觀指標

- **語音客服、自然語言理解**
 - 真人比對基準
 - 操作時間
 - 客訴比例
 - 滿意度問卷法
 - 業績成長
 - 其它

訓練調參 — 資料集分配

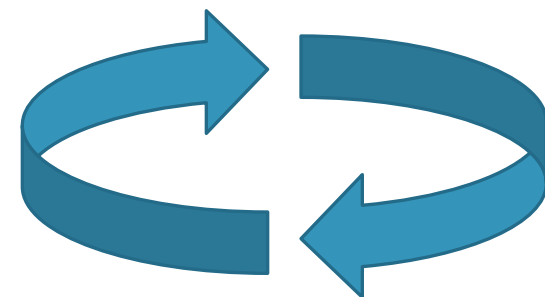
已標註之資料可依一定比例分配到訓練集、驗證集及測試集中。**絕對不可把測試集加入訓練集或驗證集中。**

分配比例**沒有一定標準**，小規模資料集建議**訓練:驗證:測試**為**6:2:2**。大規模（數百萬筆）資料集，驗證、測試各留數萬筆即可。

訓練集：就像在學校學習，了解問題，找出學習重點。

驗證集：就像模擬考，測試學習成果，找出修正方向。

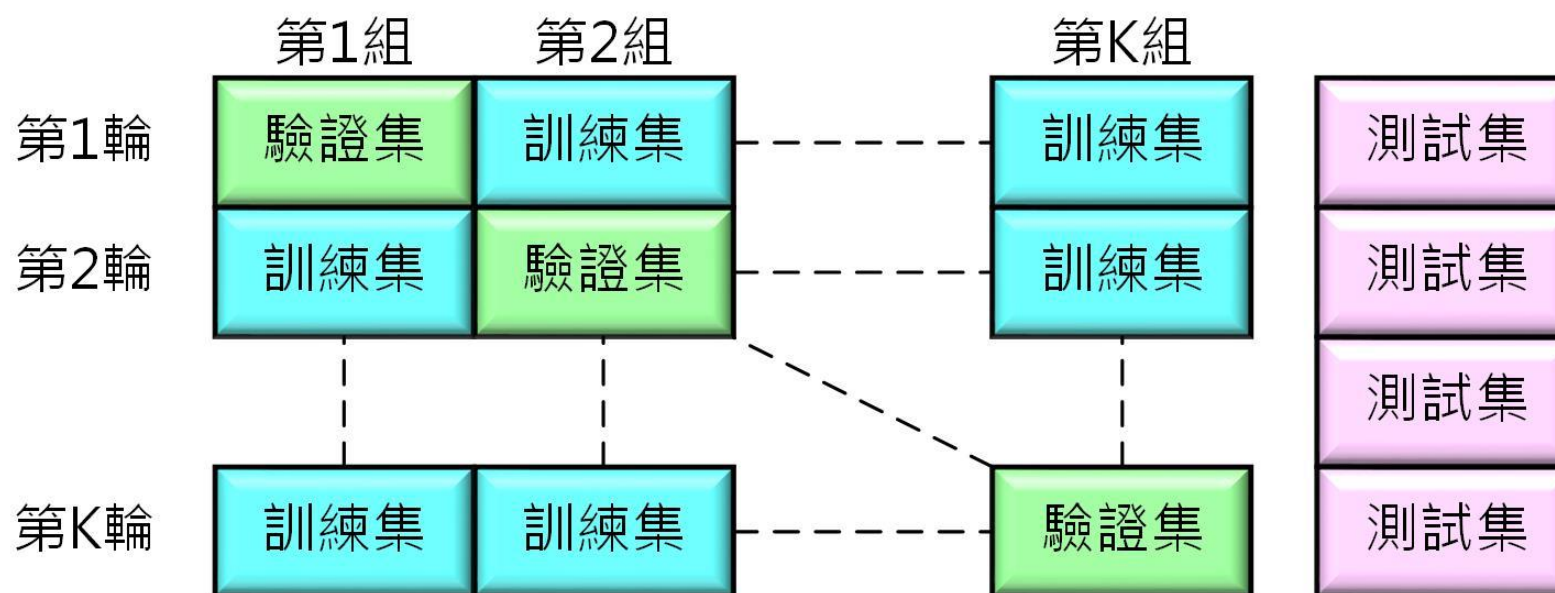
測試集：就像大考，出題和模擬考接近程度會影響最後成績。



訓練調參 — 交叉驗證



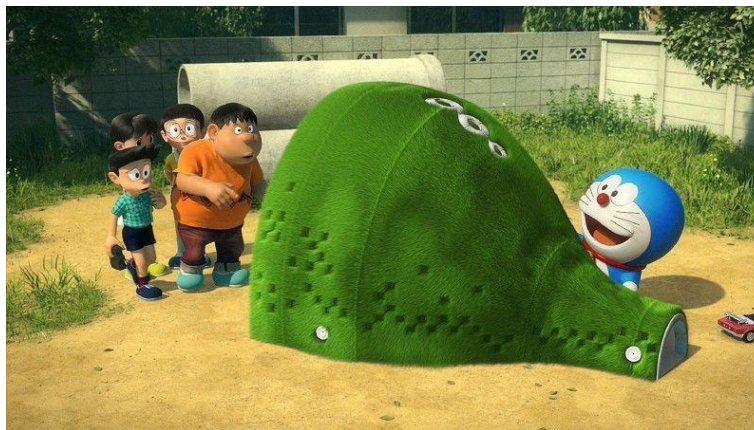
(a) 一般非重疊訓練、驗證及測試



(b) K輪交叉訓練、驗證及測試

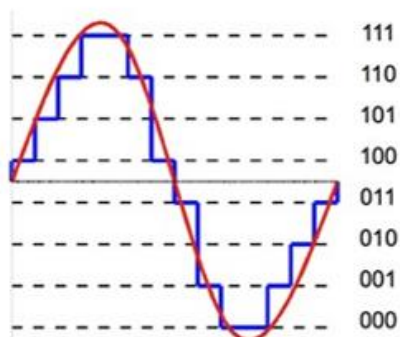
佈署推論 — 模型優化

多啦A夢的
縮小隧道



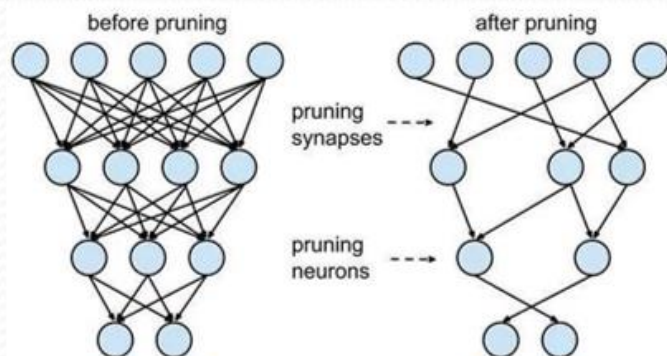
人 / 物縮小，
機能不變。

數值量化



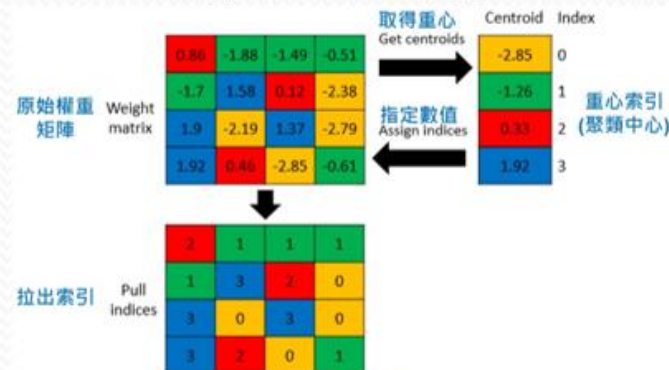
FP32 → INT8

模型剪枝



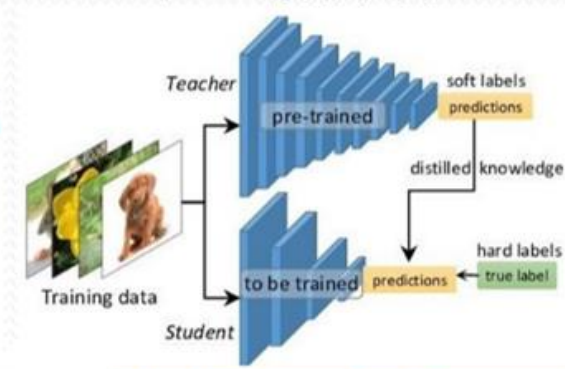
Before → After

模型壓縮



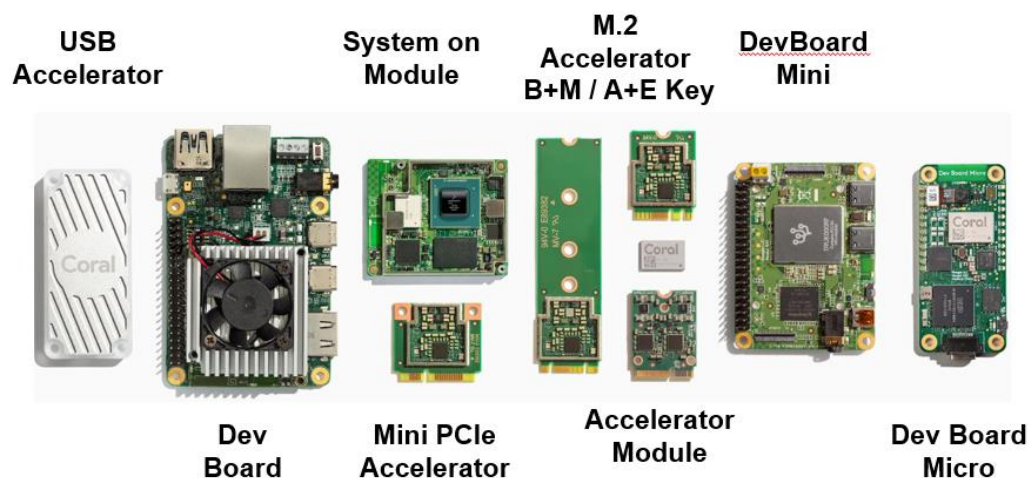
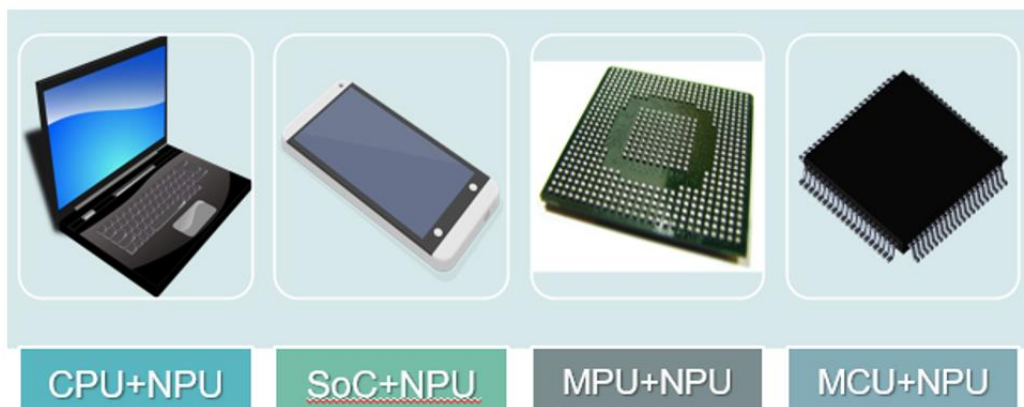
權重共享 ...

知識蒸餾



老師 → 學生

佈署推論 — 硬體選用



佈署考量重點

- 硬體成本
- 記憶體大小 (模型參數量)
- 推論速度 (FPS) (含優化處理)
- 推論精度 (FP32 / INT8)
- 待機/推論功耗 (W) (含主動散熱)
- I/O擴展性
- 異質整合介面
- **更新及誤報成本**

4.2. 開源訓練工具



- Google Colab
- Jupyter Notebook
- TensorFlow (Keras)
- PyTorch (Torch, Caffe2)
- ONNX
- OpenCV

開源訓練工具 — 免費工具



Google Colab



Google Drive

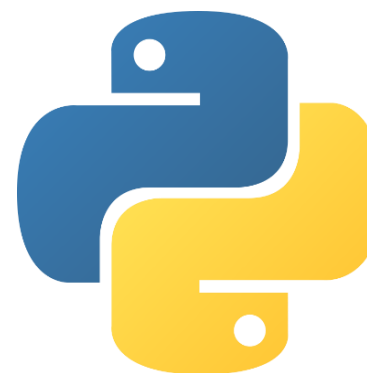
PYTORCH



TensorFlow



Jupyter Notebook



Python

Numpy
PIL
OpenCV
...

Google Colab — 基本介紹



免費虛擬機

- Xeon 2.2GHz CPU * 2
- 12GB 記憶體
- 50GB 以上儲存空間
- 免費 GPU : Nvidia T4 / Google TPU
- 付費 GPU : Nvidia A100 / V100

作業環境及預裝套件

- Liunx (Ubuntu)
- Python
- Jupyter notebook
- TensorFlow
- PyTorch
- OpenCV

額外套件(線上安裝)

- Keras, MxNet, XGBoost....

使用限制

- 最多連續12小時(含額外安裝)
- 容易連線中斷

Google Colab — 啟動



Google Colab — 設定硬體加速器

Jupyter Notebook 工作模式



The screenshot shows the Google Colab interface in Jupyter Notebook work mode. The top bar includes the Google Colab logo, the file name 'Untitled1.ipynb', and a '檔案名稱' (File Name) label. The main menu bar contains '檔案' (File), '編輯' (Edit), '檢視畫面' (View), '插入' (Insert), '執行階段' (Runtime), '工具' (Tools), '說明' (Help), and '已儲存所有變更' (Save all changes). The '編輯' (Edit) menu is open, showing options like '復原插入儲存格' (Undo), '重做' (Redo), '選取所有儲存格' (Select all), '剪下儲存格或選取範圍' (Cut), '複製儲存格或選取範圍' (Copy), '貼上' (Paste), '刪除所選儲存格' (Delete), '尋找與取代' (Find and replace), '尋找下一個項目' (Find next), '尋找上一個項目' (Find previous), '筆記本設定' (Notebook settings), '顯示/隱藏程式碼' (Show/Hide code), and '清除所有輸出內容' (Clear all outputs). The '筆記本設定' (Notebook settings) dialog is open, showing the '執行階段類型' (Runtime type) set to 'Python 3'. Under '硬體加速器' (Hardware accelerator), the 'CPU' option is selected, and the 'T4 GPU' option is highlighted with a blue arrow. The '想要使用付費 GPU 嗎?' (Do you want to use a paid GPU?) section is visible, with a link to '購買額外運算單元' (Purchase additional compute units). The 'RAM' and '磁碟' (Disk) usage is shown as 0% in the top right corner.

Google Colab — 檢查硬體配置 (選配)

```
print("CPU Status:")
```

```
!cat /proc/cpuinfo | grep 'model name' #檢查CPU資訊
```

```
print("\nDisk Status:")
```

```
!df -lh #檢查磁碟空間
```

```
print("\nRAM Status:")
```

```
!free -h #檢查記憶體大小
```

```
model name      : Intel(R) Xeon(R) CPU @ 2.20GHz
model name      : Intel(R) Xeon(R) CPU @ 2.20GHz
Filesystem      Size  Used Avail Use% Mounted on
overlay         108G   25G   84G   23% /
tmpfs           64M    0    64M    0% /dev
shm             5.8G    0   5.8G    0% /dev/shm
/dev/root       2.0G   1.1G   849M   57% /usr/sbin/docker-init
tmpfs           6.4G  120K   6.4G    1% /var/colab
/dev/sda1       70G   43G   27G   62% /kaggle/input
tmpfs           6.4G    0   6.4G    0% /proc/acpi
tmpfs           6.4G    0   6.4G    0% /proc/scsi
tmpfs           6.4G    0   6.4G    0% /sys/firmware

              total        used        free      shared  buff/cache   available
Mem:          12Gi         556Mi        8.4Gi         1.0Mi         3.8Gi         11Gi
Swap:          0B           0B           0B
```

Google Colab — 檢查GPU配置 (選配)

筆記本設定須選定 **Nvidia GPU (T4 / A100 / V100)**

!nvidia-smi #檢查NVIDIA GPU使用狀態及參數

```
+-----+
| NVIDIA-SMI 535.104.05                Driver Version: 535.104.05   CUDA Version: 12.2     |
+-----+-----+-----+-----+-----+-----+
| GPU  Name           Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|                                           | MIG M.         |
+=====+=====+=====+=====+=====+=====+
|   0   Tesla T4               Off  | 00000000:00:04.0 Off |                    0 |
| N/A   43C    P8              9W / 70W |  0MiB / 15360MiB |      0%      Default |
|                                           | N/A             |
+-----+-----+-----+-----+-----+-----+

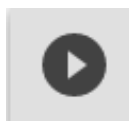
+-----+
| Processes:                                             |
|  GPU   GI    CI          PID    Type   Process name                      GPU Memory |
|          ID    ID                                   |          Usage  |
+=====+=====+=====+=====+=====+=====+
| No running processes found                            |
+-----+
```


Google Colab — 上傳檔案執行

上傳檔案

- 將 *.ipynb 上傳到Google 雲端硬碟，再以Colab開啟、執行。

執行程式



- 直接於編輯區中輸入程式，可單步執行（按  或 **Ctrl + Enter**）或全部一起執行。

執行階段	工具	說明	已儲存所有變更
全部執行		Ctrl+F9	
執行上方的儲存格		Ctrl+F8	
執行聚焦的儲存格	單步執行	Ctrl+Enter	
執行選取範圍		Ctrl+Shift+Enter	
執行下方的儲存格		Ctrl+F10	
中斷執行		Ctrl+M	
重新啟動執行階段		Ctrl+M .	
重新啟動並執行所有儲存格			
恢復原廠設定的執行階段			
變更執行階段類型			
管理工作階段			
查看執行階段記錄			

Jupyter Notebook — 主畫面



編寫完程式要按【檔案】－【儲存】儲存目前編輯內容到雲端硬碟。
檔案亦可下載到本地端，格式可選 *.ipynb 或 *.py

Jupyter Notebook — Hello World

1.

滑鼠移動最上方，出現橫線，按【+文字】，插入一行文字欄位。



2.

姓名：Jack Hsu 學號：12345678

按兩下 (或按 Enter 鍵) 即可編輯。

3.

`print('Hello World!')`

輸入程式碼 `print('Hello World!')`

4.

`print('Hello World!')`
Hello World!

按下單步執行鍵或按【Ctrl+Enter】執行程式，即會產生對應結果。
按【Shift+Enter】執行後自動跳到下一行。

Jupyter Notebook — 運行 Github 範例

- 要先於瀏覽器中登錄Google帳號才使用Colab
- 一般於Github中 *.ipynb 完整路徑名稱多半為

https://github.com/使用者名稱/專案名稱/blob/main/子目錄名稱/*.ipynb

- 將其改成下列型式即可直接啟動運行

https://colab.research.google.com/github/使用者名稱/專案名稱/blob/main/子目錄名稱/*.ipynb

Jupyter Notebook — Magic Command

在Colab環境下加上！即可執行Linux命令

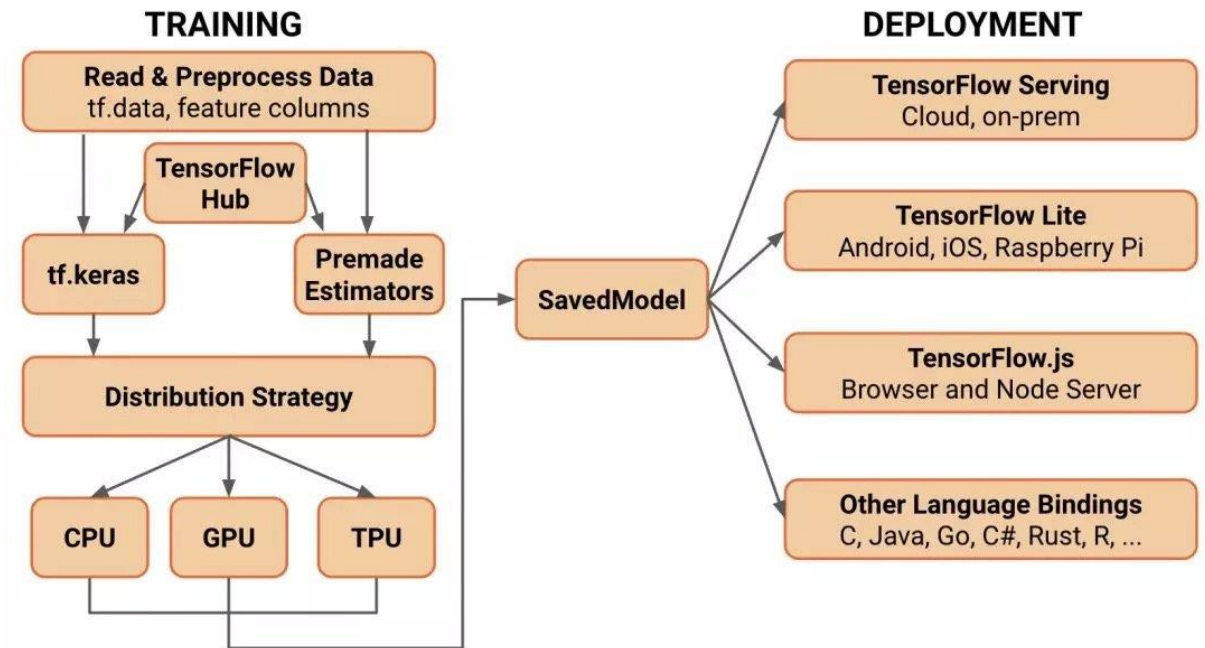
- **! ls** ，查詢當下目錄內容，相當於Windows下dir指令
- **! pwd** ，顯示目前工作目錄
- **! rm xxx.xxx** ，刪除檔案，加**-r**為刪除路徑下所有檔案（**危險**）
- **%cd** 路徑名稱 ，切換工作路徑
- **%pycat xxx.py** ，開啟python程式原始碼

TensorFlow — 簡介

- **開發者**：Google Brain
- **首次發佈**：2015/1/9
- **程式語言**：Python, C++, CUDA
- **支援平台**：Linux, macOS, Windows, Android
- **主要功能**：機器學習及深度學習模型開發、訓練及部署



- 1.x 和 2.x 版本有很大差異。
- 2.x 已將 **Keras** 納入。
- 2.x 訓練及部署架構如下所示。



資料來源：<https://zh.wikipedia.org/zh-tw/TensorFlow>

TensorFlow — Model Garden



TensorFlow Model Garden

將各種已預訓練模型置於 Github 上，
方便應用到不同裝置。



TensorFlow Hub

將各種已預訓練模型置於 Kaggle 上，
方便應用到不同裝置。

電腦視覺

影像分類、物件偵測、
影像分割、影片分類

自然語言處理

大語言模型、機器翻譯、
文字生成、知識蒸餾

推薦系統

DLRM, DCNv2, NCF

[https://github.com/tensorflow/
models/tree/master/official](https://github.com/tensorflow/models/tree/master/official)

[https://www.kaggle.com/models
?tfhub-redirect=true](https://www.kaggle.com/models?tfhub-redirect=true)

TensorFlow — Datasets

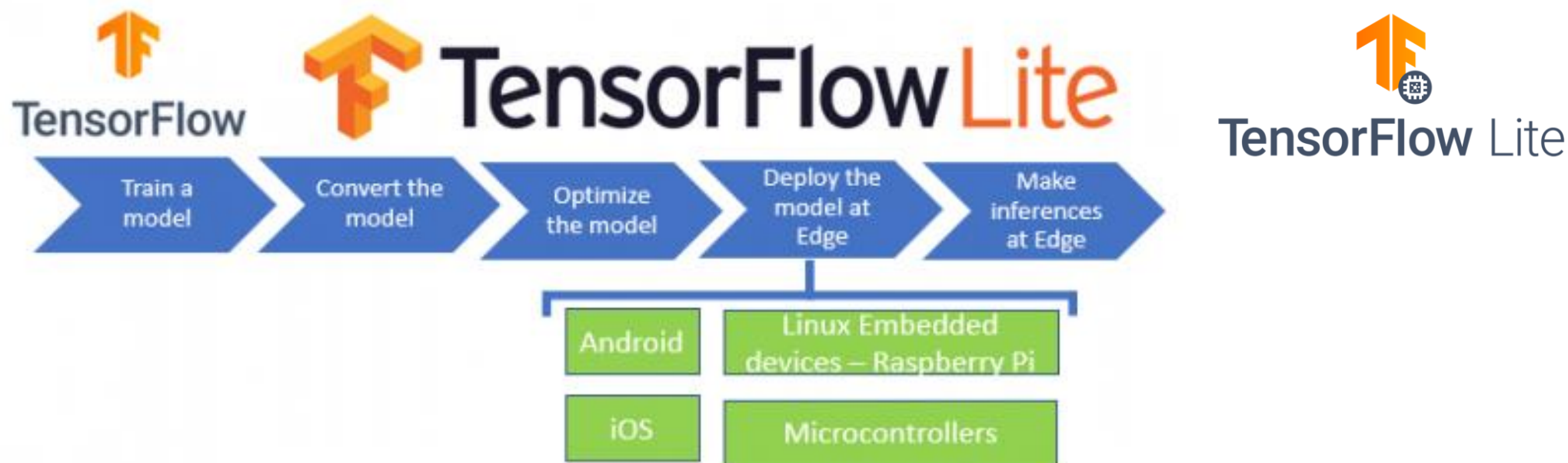


提供近八十類資料集可供參考，領域包括影像、視訊、聲音、自然語言、文字等，可直接用於模型訓練。

Overview					
Dataset Collections	Density estimation	Image segmentation	Object detection	Scene classification	Text classification
3d	Dependency parsing	Image super resolution	Open domain question answering	Semantic segmentation	Text classification toxicity prediction
Abstractive text summarization	Dialog act labeling	Image to image translation	Out of distribution detection	Sentiment analysis	Text generation
Anomaly detection	Dialogue	Instance segmentation	Question answering	Sequence modeling	Text simplification
Audio	Document summarization	Language modeling	Question generation	Sequence to sequence language modeling	Text summarization
Biology	Fine grained image classification	Linguistic acceptability	Ranking	Speech	Time series
Categorical	Graph	Machine translation	Reading comprehension	Speech recognition	Token classification
Common sense reasoning	Graphs	Monolingual	Recommendation	Structured	Tracking
Computer science	Image	Movies and tv shows	Reinforcement learning	Summarization	Trajectory
Conditional image generation	Image classification	Multilingual	Rgb d	Table to text generation	Translate
Coreference resolution	Image clustering	Natural language inference	RI unplugged	Tabular	Uncategorized
D4rl	Image compression	Natural language understanding	Rlds	Text	Unsupervised anomaly detection
	Image generation	Nearest neighbors	Robotics		Video

<https://www.tensorflow.org/datasets/catalog/overview>

TensorFlow — 衍生家族



TensorFlow Lite (TFLite)

手機、平板 (Android, iOS) 、
筆電、單板微電腦 (Linux, Pi ...)

<https://www.tensorflow.org/lite>

TensorFlow Lite for Microcontroller (TFLu)

單晶片 (MCU, EdgeTPU, Mbed...)

<https://www.tensorflow.org/lite/microcontrollers>

PyTorch — 簡介

- **原作者**：Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan
- **初始版本**：2016年10月
- **授權方式**：BSD
- **作業系統**：Linux, macOS, Windows
- **程式語言**：Python, C++, CUDA
- **PyTorch**是一個開源的**Python**機器學習庫，基於**Torch**，底層由C++實現，應用於人工智慧領域。
- 最初由Facebook團隊研發，2018年3月將**Caffe2**合併。
- 兩大特徵
 - 類似**NumPy**的張量計算，可用GPU加速。
 - 基於帶自動微分系統的深度神經網路。



資料來源：<https://zh.wikipedia.org/wiki/PyTorch>

PyTorch — 主要模組

Python API

- torch.nn
- torch.Tensor
- torch.autograd
- torch.cuda
- torch.backends
- torch.distributed
- torch.distributions
- torch.fft
- torch.futures
- torch.fx

- torch.hub
- torch.jit
- torch.linalg
- torch.overrides
- torch.profiler
- torch.onnx
- torch.optim
- torch.random
- torch.sparse
- torch.Storage
- torch.utils

Libraies

- torchaudio
- torchtext
- **torchvision**
- TorchElastic
- TorchServe
- PyTorch on XLA Devices

資料來源：<https://pytorch.org/docs/stable/index.html>

Pytorch — 支援資料集

torchvision.datasets

- CelebA
- **CIFAR**
- Cityscapes
- **COCO**
- DatasetFolder
- EMNIST
- FakeData
- Fashion-MNIST
- FlickrHM
- DB51
- ImageFolder
- **ImageNet**
- Kinetics-400
- KMNIST
- LSUN
- **MNIST**
- Omniglot
- PhotoTour
- Places365
- QMNIST
- SBD
- SBU
- STL10
- SVHN
- UCF101
- USPS
- **VOC**

資料來源：<https://pytorch.org/vision/stable/models.html>

Pytorch — 支援模型

【影像分類】

torchvision.models

- AlexNet
- VGG
- ResNet
- SqueezeNet
- DenseNet
- Inception v3
- GoogLeNet
- ResNeXt

➤ ShuffleNet v2

➤ MobileNet v2

➤ Wide

➤ ResNet

➤ MNASNet

【語義分割類】

torchvision.models

- FCN ResNet50/101
- DeepLabV3 ResNet50/101
- DeepLabV3 MobileNetV3-Large
- LR-ASSP MobileNetV3-Large

【人體關鍵點類】

torchvision.models

- Faster R-CNN ResNet-50 FPN
- MaskR-CNN ResNet-50 FPN
- RetinaNet RstNet-50 FPN

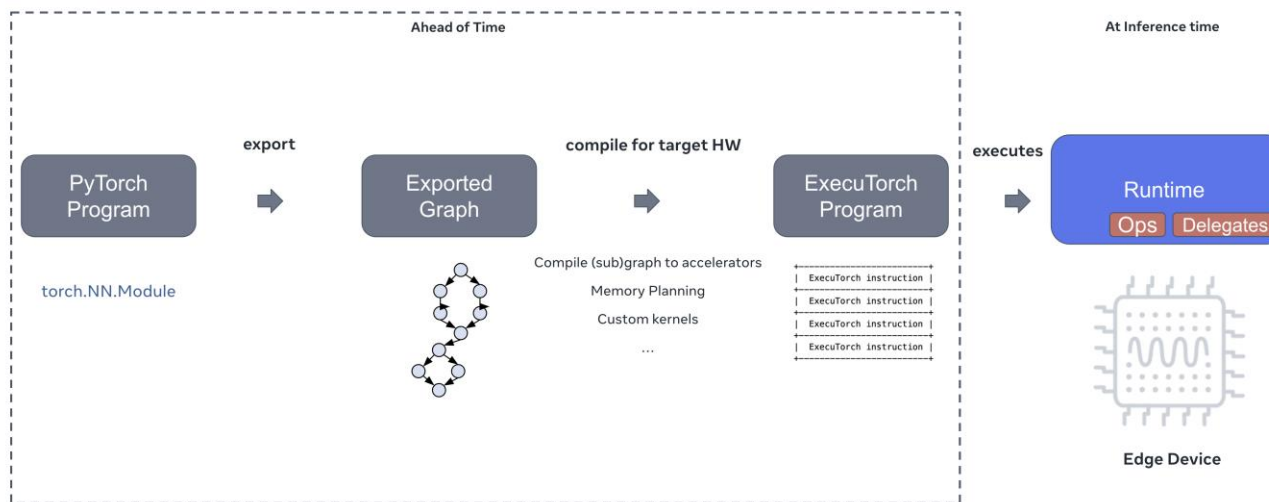
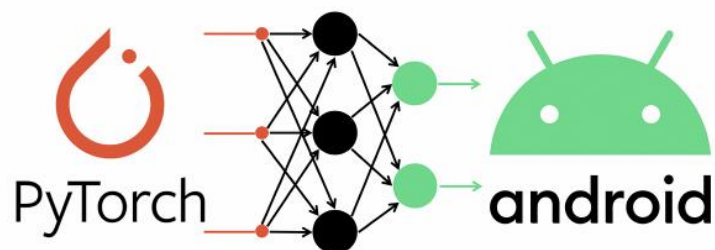
【影片分類】

torchvision.models

- ResNet 3D 18
- ResNet MC 18
- ResNet (2+1)D

資料來源：<https://pytorch.org/vision/stable/models.html>

PyTorch — 衍生家族



PyTorch Mobile (Live)

行動裝置使用

<https://pytorch.org/mobile/home/>

ExecuTorch

邊緣裝置及單晶片使用

<https://pytorch.org/executorch-overview>



ONNX — 簡介

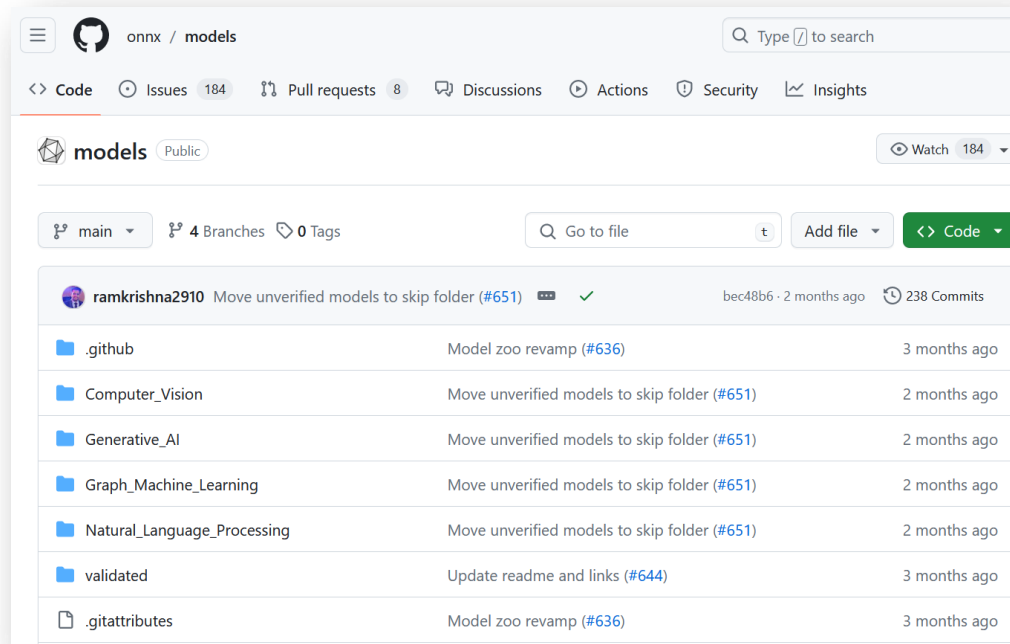


資料來源：<https://onnx.ai/>

ONNX (Open Neural Network Exchange)

- 是一種針對機器學習所設計的開放式的文件格式，用於存儲訓練好的模型。相當於AI文件的PDF格式。
- 由Facebook, Microsoft, Amazon, IBM共同推動。
- 可支援各種框架互相交換，如Caffe2, PyTorch, MXNet, CNTK, NL.NET, TensorFlow, Keras。
- 可作為硬體抽象層。

ONNX — Model Zoo



- 影像分類
- 物件偵測
- 影像分割
- 人臉/身體/手勢分析
- 影像操作（超解析度、風格轉移...）
- 語音和聲音處理
- 機器理解（自然語言）
- 機器翻譯
- 視覺問答

資料來源：<https://github.com/onnx/models>

OpenCV — 簡介

原創作者：Intel

初始版本：2000年6月

正式版本：2006年(V1.0)

授權方式：BSD授權條款

作業系統：Window, Liunx, Mac, iOS, Android ...

開發語言：C, C++, C#, Java, Python, Matlab, Ruby ...

硬體加速：TBB, IPP, CUDA, OpenCL ...

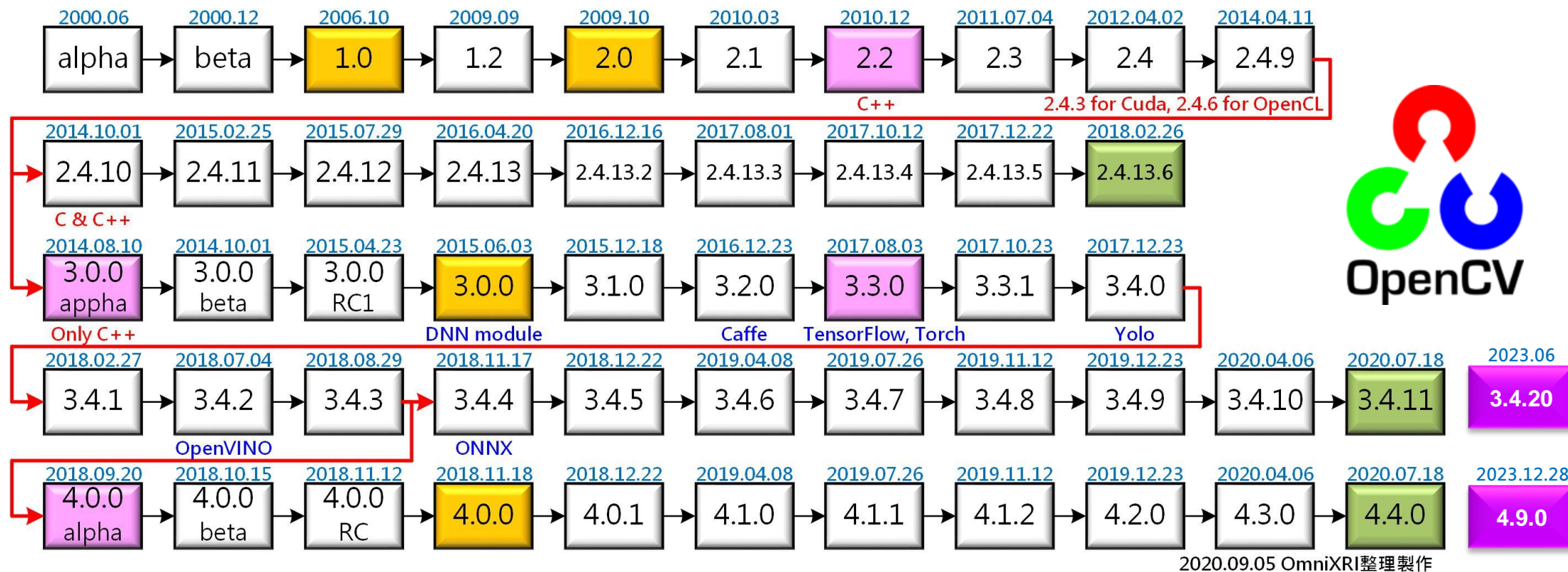
主要目標：

- 為推進機器視覺的研究，提供一套開源且最佳化的基礎庫。**不重造輪子。**
- 提供一個共同的基礎庫，使得開發人員的代碼更容易閱讀和轉讓，促進了知識的傳播。
- 提供一個**開源、免費**的軟體授權，促進商業應用軟體的開發。



資料來源：<https://zh.wikipedia.org/wiki/OpenCV>

OpenCV — 演進歷史



第三方開發工具：

Emgu CV

語言：.NET C#, VB, VC++, Python ...

平台：Windows, Linux, Mac OS, iOS, Android ...

版本：4.3.0 (2020.06.07)

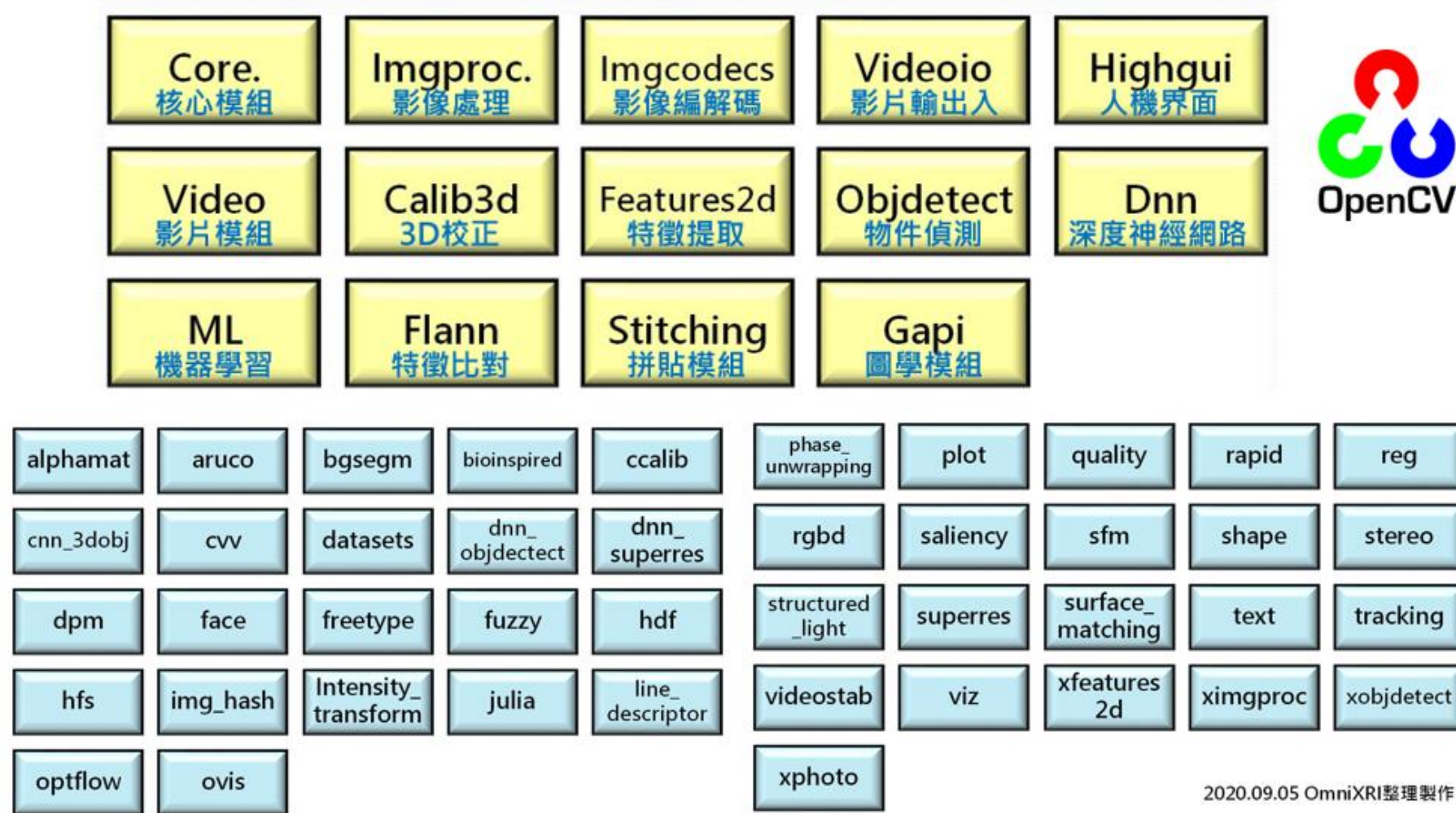
OpenCV for Unity

語言：Java,

平台：iOS, Android, Win10, WebGL, Linux

版本：OpenCV 4.4.0 Assets Plugin

OpenCV — 主要模組(4.4.0)



OpenCV — DNN模組

深度神經網路模組（直接讀取模型）

[cv::dnn::readNetFromCaffe](#) (3.2.0 ↑)

[cv::dnn::readNetFromTensorflow](#) (3.3.0 ↑)

[cv::dnn::readNetFromTorch](#) (PyTorch, 3.3.0 ↑)

[cv::dnn::readNetFromDarknet](#) (YOLO, 3.4.0 ↑)

[cv::dnn::readNetFromModelOptimizer](#) (Intel OpenVINO IR, 3.4.2 ↑)

[cv::dnn::readNetFromONNX](#) (3.4.4 ↑)

[cv::dnn::readNetFromTFLite](#) (4.7.0 ↑)



資料來源：https://docs.opencv.org/4.x/d6/d0f/group__dnn.html

OpenCV — Colab顯示方式

- 在Colab上由於沒有實體顯示器可對應，所以無法使用 `cv2.imshow()` 這類指令，需改用其它方式顯示影像及影片。
- 可改用下列方式顯示
 - 透過matplotlib `show()`顯示
 - 透過IPython.display及PIL顯示
 - 以Colab自帶`cv2_imshow()`函式顯示
 - 透過IPython.display `HTML()`及`base64 b64encode()`函式庫顯示視頻
 - 利用ffmpeg處理OpenCV `VideoWriter()`產出視頻播放問題

Colab 完整範例：

https://github.com/OmniXRI/Colab_OpenCV_Display

小結

➤ AI工作流程

- 要先定義AI待解問題、輸出入關係、規畫工作流程，接著選用適當模型進行訓練，利用可視化工具和成果指標來調參，最後進行模型優化，選用合適硬體進行推論。

➤ 開源訓練工具

- 認識Google Colab, Jupyter Notebook, TensorFlow, PyTorch, ONNX, OpenCV及基本功能，有助於後續學習完整AI範例。

參考文獻

- 許哲豪，臺灣科技大學資訊工程系「人工智慧與邊緣運算實務」(2021~2023)

<https://omnixri.blogspot.com/p/ntust-edge-ai.html>

- 許哲豪，NTUST Edge AI ChD (2022) Google Colab進階應用

<https://omnixri.blogspot.com/p/ntust-edge-ai-chd-2022.html>

延伸閱讀

- 許哲豪，【課程簡報】20210326_東南資科_創新應用實務02_OpenCV初體驗

<https://omnixri.blogspot.com/2021/03/2021032602opencv.html>

- 許哲豪，【課程簡報】20210409_東南資科_創新應用實務03_OpenCV彩色影像處理

<https://omnixri.blogspot.com/2021/04/2021040903opencv.html>

- 許哲豪，如何在Google Colab上使用本地端Webcam即時運行OpenCV自定義影像處理函式

<https://omnixri.blogspot.com/2022/09/google-colabwebcamopencv.html>

- 許哲豪，如何在Colab上顯示雲端硬碟(Google Drive)上的影像和視頻

<https://omnixri.blogspot.com/2020/12/colabgoogle-drive.html>

沒有最邊



只有更邊

歡迎加入
邊緣人俱樂部



歐尼克斯實境互動工作室
(OmniXRI Studio)

許哲豪 (Jack Hsu)

Facebook : Jack Omnixri

FB社團 : Edge AI Taiwan邊緣智能交流區

電子信箱 : omnixri@gmail.com

部落格 : <https://omnixri.blogspot.tw>

開 源 : <https://github.com/OmniXRI>

YOUTUBE 直播 : <https://www.youtube.com/@omnixri1784/streams>