

Core ML 相關 Sessions

濃縮呈現

Marvin Lin

<https://github.com/MoonAndEye/CoreMLDemo>

和 Core ML 相關的 Sessions

- WWDC18 中，和 Core ML 有關的 Session

703 - Introduction Create ML 34 min

708 - What's new in Core ML - Part 1 37 min

709 - What's new in Core ML - Part 2 34 min

712 - A guide to Turi Create 36 min

713 - Introducing natural language framework 38 min

716 - Object tracking in vision 34 min

717 - Vision with core ML 39 min

Total: 252 min

無法在這個 talk 放入的內容

Session 703 - CreateML

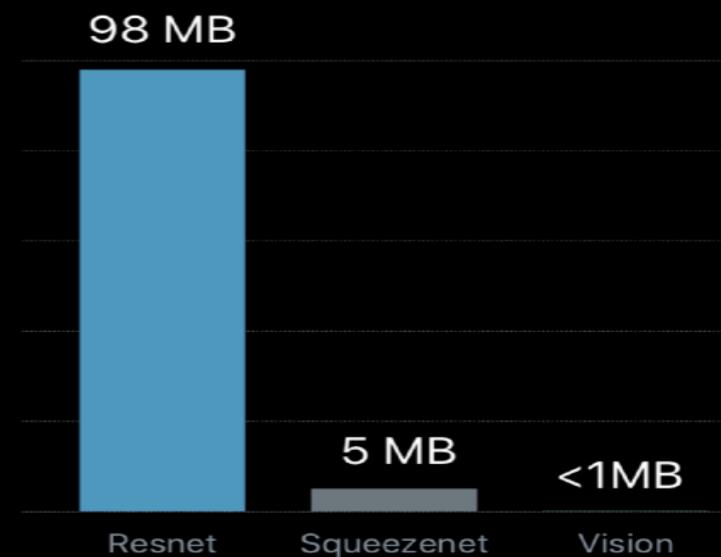
使用 Playground 訓練模型 (下一個 talk 的講者在這個場地會介紹)，如果是 Image Classification 的任務，建議用 Create ML

Session 708 & 709 - What's new in Core ML (部分內容)

Quantization (縮小模型的容量)

batch processing (提升批次處理訓練的速度)

custom layer in CNN (導入最新理論的 layer 在繼有模型上).



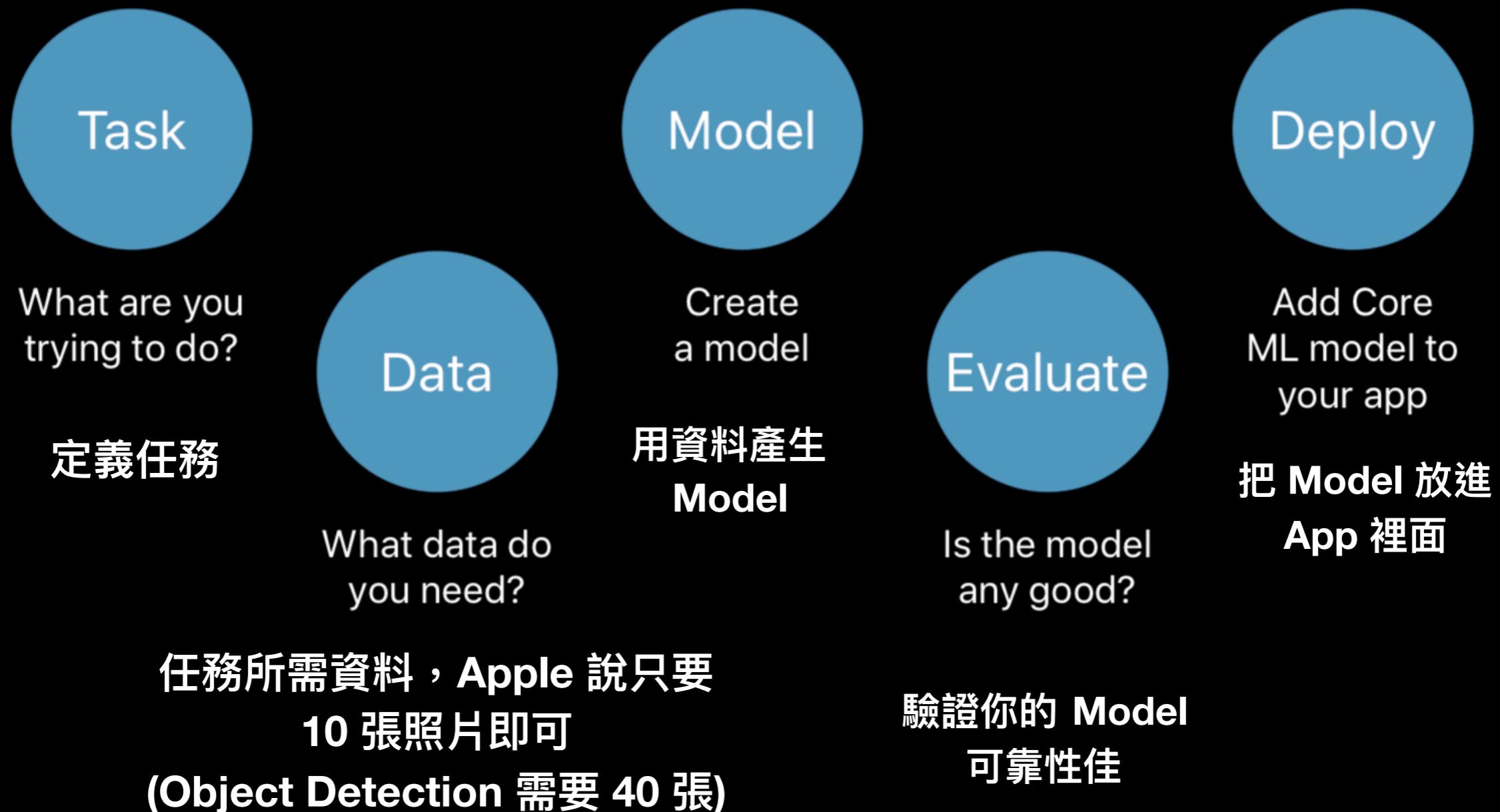
Session 716 & 717 - Vision (部分內容)

VNTranslationallImageRegistrationRequest (比對畫面移動,減少硬體消耗)

Tracking Object (追蹤的演算法比較輕量，且畫面呈現比較滑順)

用結構化資料訓練模型

要做 ML Model 所需要的五個基本步驟



App Demo

- 金城武判定器
- 圖案濾鏡 Style Transfer - 克林姆風格、莫內風格、梵谷風格 etc...
- Object Detection - 威利在哪裡？

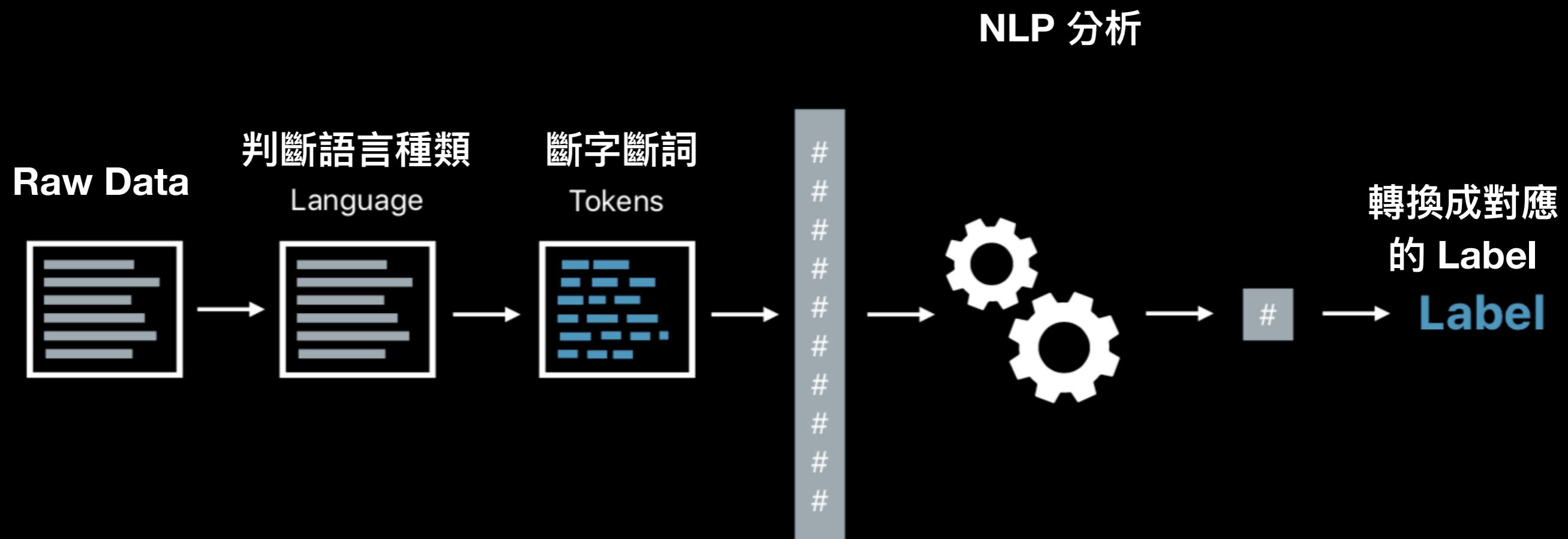
<https://github.com/MoonAndEye/CoreMLDemo>



Simple model creation
Tailored to your app
Leverages core Apple technologies
Powered by Mac



基本的自然語言分析流程



NLP - 辨試語系

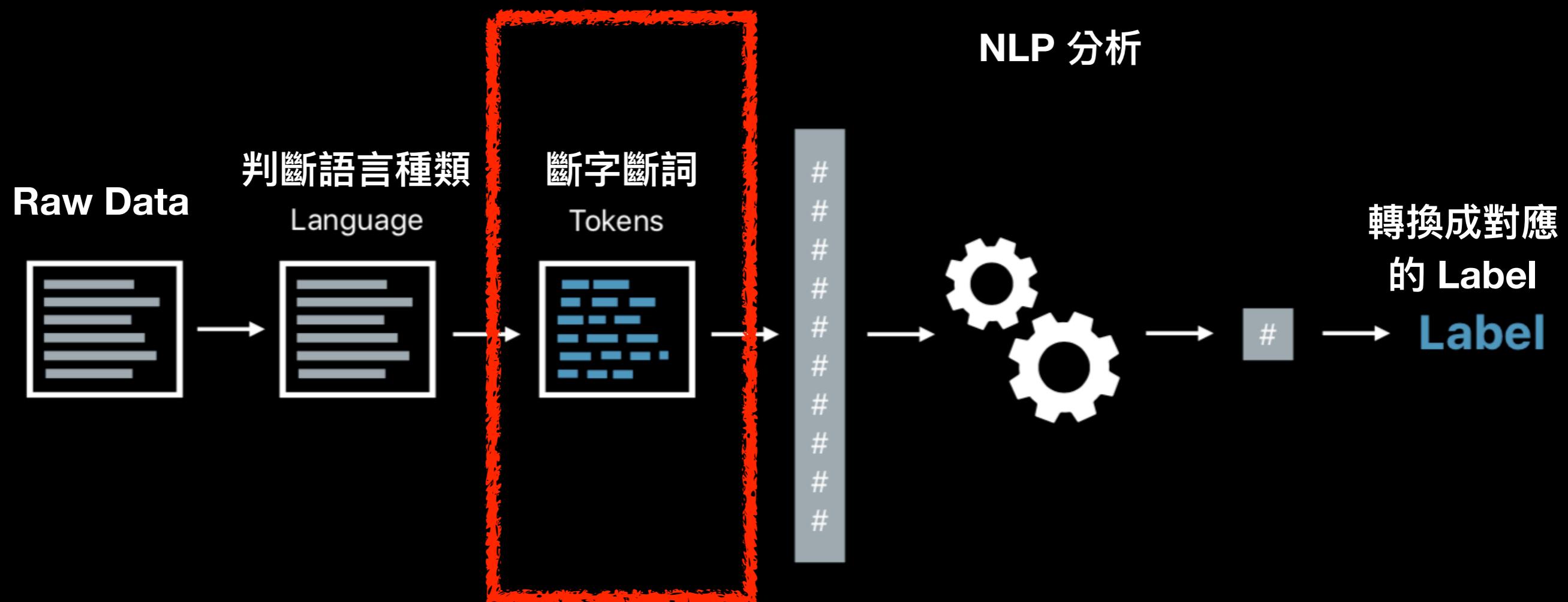
```
let recognizer = NLLanguageRecognizer()  
let testStr = "今日的臺北市只包含12個區，面積為271.7997平方公里"  
recognizer.processString(testStr)  
let lang = recognizer.dominantLanguage  
let hypothesis = recognizer.languageHypotheses(withMaximum: 4)|  
  
▶({_rawValue "zh-Hans"}, value 0.0001164831774076447)  
▶({_rawValue "zh-Hant"}, value 0.994546115398407)  
▶({_rawValue "ja"}, value 0.005327472928911448)
```

語系碼使用 ISO 639-1，JA 表示日文

歐洲語系沒有問題，但是馬來語和印尼語無法正確判斷

```
let recognizer = NLLanguageRecognizer()  
let testStr = "sila berhati hati ruang di platform"  
recognizer.processString(testStr)  
let lang = recognizer.dominantLanguage  
let hypothesis = recognizer.languageHypotheses(withMaximum: 4)|  
  
▶({_rawValue "it"}, value 0.9825760722160339)  
▶({_rawValue "hr"}, value 0.007782762870192528)  
▶({_rawValue "de"}, value 0.000978454714640975)  
▶({_rawValue "tr"}, value 0.005278028082102537)
```

基本的自然語言分析流程



全語系斷詞系統 - NLTokenizer

```
43 // Document, paragraph, sentence, word
44 let tokenizer = NLTokenizer(unit: .word)
45 let str = "下一個講者的演講一定很精采，對機器學習有興趣的朋友請留下來聽"
46 let strRange = str.startIndex ..< str.endIndex
47 tokenizer.string = str
48 // tokens 會輸出 Range<Set>,
49 let tokenArray = tokenizer.tokens(for: strRange)
50 for each in tokenArray {
51     print(each)
}
43 // Document, paragraph, sentence, word
44 let tokenizer = NLTokenizer(unit: .word)
45 let str = "この戦争が終わったら、俺は故郷に帰って結婚するんだ"
46 let strRange = str.startIndex ..< str.endIndex
47 tokenizer.string = str
```

下一個講者的演講一定很精采，對機器學習有興趣的朋友請留下來聽

きょう
の
ぼく
に
げんかい
は
ない

この
戦争
が
終
わっ
たら
俺
は
故郷
に
帰っ
て
結婚
する
んだ

目前知名的 ML 框架

IBM Watson®

turi 

 ONNX

 Yandex
CatBoost

^{dmk}
XGBoost

K Keras

Caffe

 scikit
learn

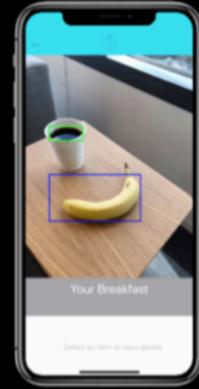
LIBSVM

mxnet

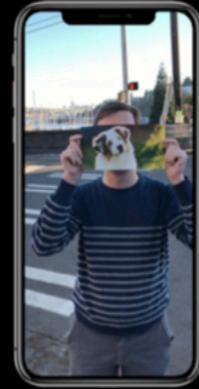
Turi Create 是什麼？

- 他是一個 Python 的 Package，可以製作 Core ML models，而且有開源
- 簡單易用，即使非 ML 專門也可以操作
- 讓 iOS 開發者只需要專注在 APIs 上
- 跨平台 (Mac, Linux)
- 學習成本低
 - pip install turicreate (Mac 已經內建 Python 2.7)
 - 低於 20 行的程式碼
 - Jupyter Notebook (非必要)

<https://github.com/apple/turicreate>



偵測並辨視物體



即時偵測



手勢操作



推薦個性化鬍子

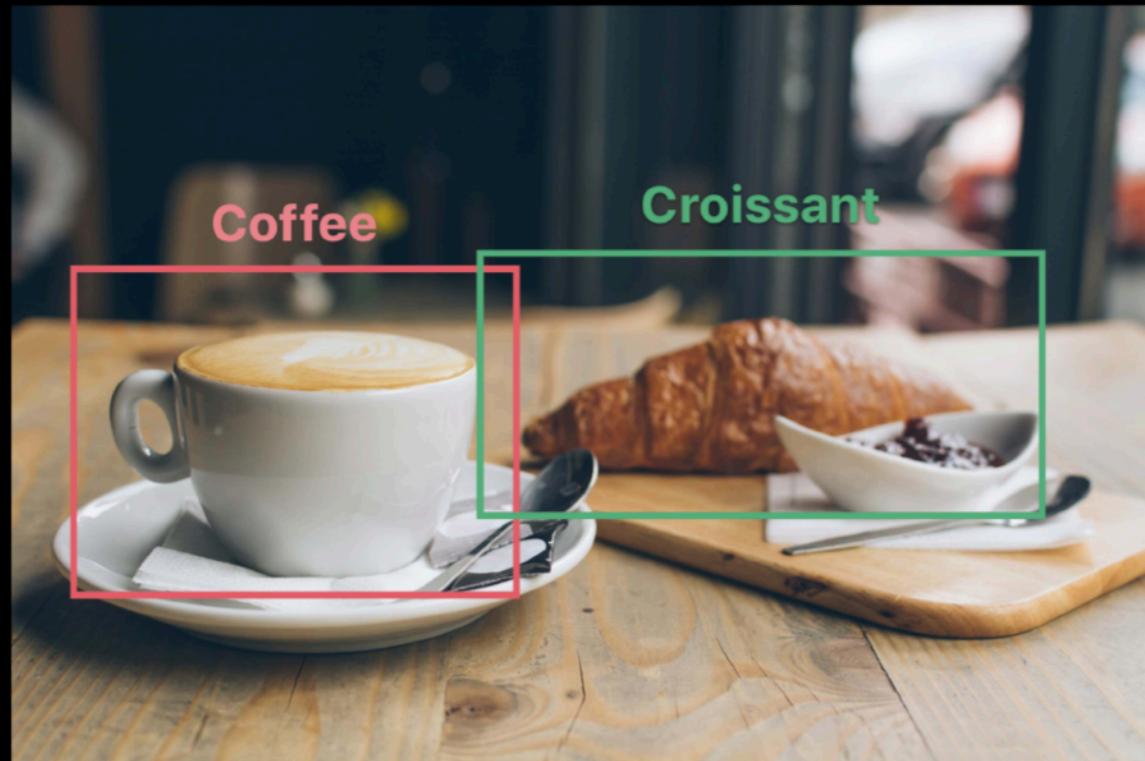


風格變換

Jupyter Notebook

- Image Classifier
- Style Transfer
- Object Detection

Object Detection

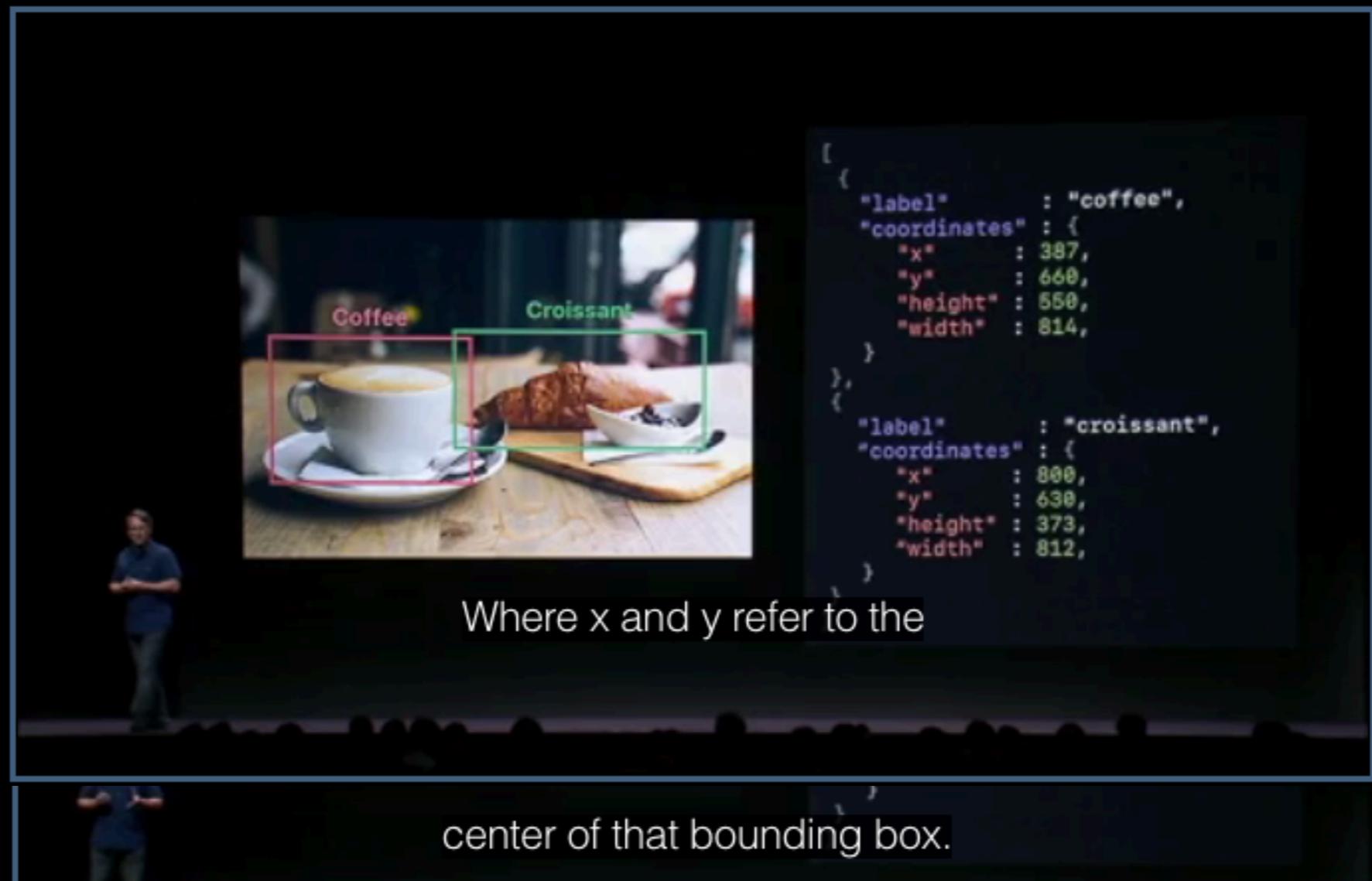


```
[  
  {  
    "label"      : "coffee",  
    "coordinates": {  
      "x"   : 387,  
      "y"   : 660,  
      "height": 550,  
      "width" : 814,  
    }  
  },  
  {  
    "label"      : "croissant",  
    "coordinates": {  
      "x"   : 800,  
      "y"   : 630,  
      "height": 373,  
      "width" : 812,  
    }  
  }  
]
```

標記會有個 label 和座標

要注意的是, x , y 在 turi create 裡面是中心點

你的 X, Y 是我的 X, Y 嘽?



要注意 x , y 在 turi create 裡面是中心點

要注意 x , y 在 turi create 裡面是中心點

要注意 x , y 在 turi create 裡面是中心點

Vision RecognizedObject 的座標

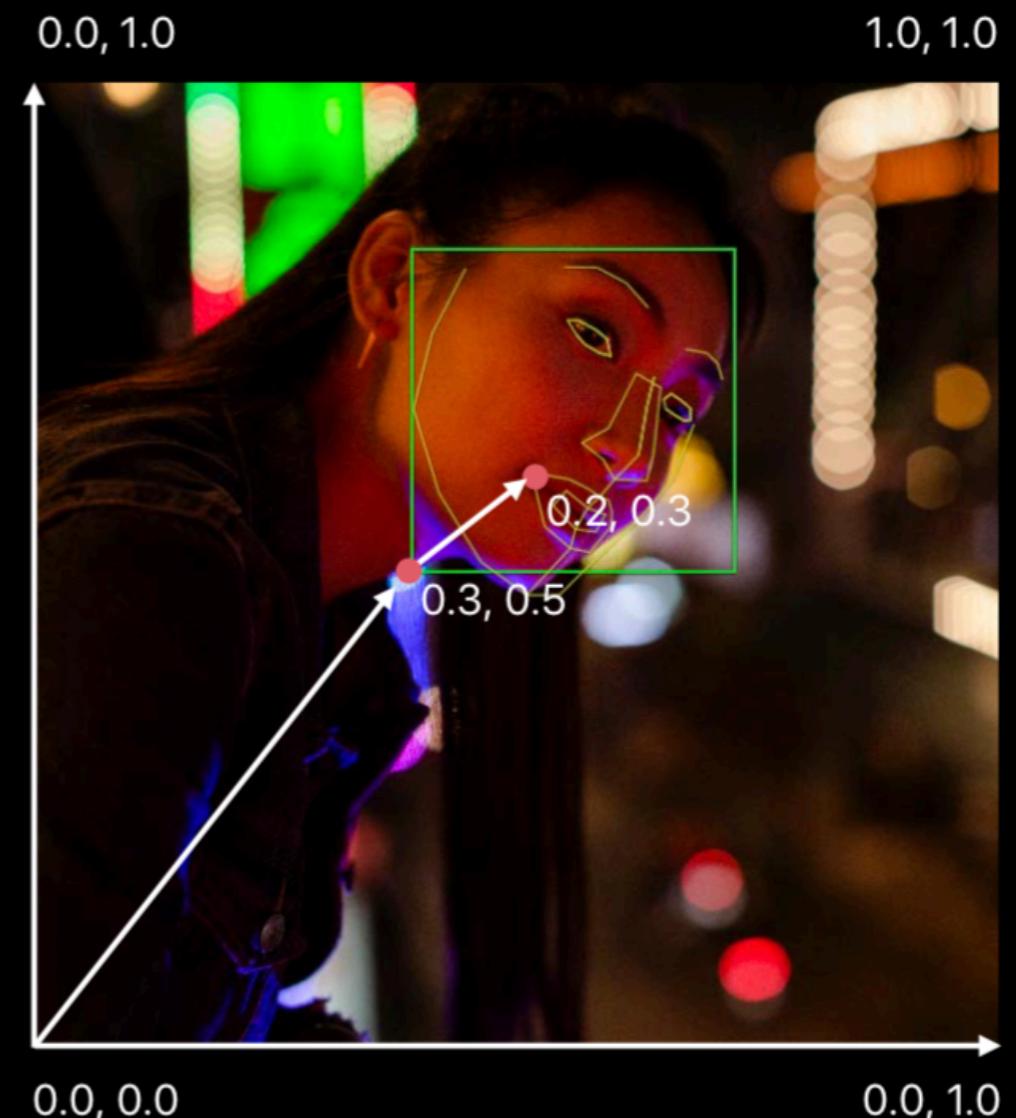
原點在左下角

座標系統 Normalized

VNUtils.h 有提供座標系統轉換工具

(但我轉出來後 frame 有偏移，不知道問題在 model 還是座標轉移)

使用 VNRequest 的時候，
imageCropAndScaleOption 建議
設定為 .scaleFill 比較準



如果是 Object Detect 要驗證一下模型

標記是不是正確的
範圍是不是正確的



正確的標記範例



Overlap 10%



Overlap 70%



Overlap 99%



Need **correct labels** AND **at least 50%** overlap in boxes

需要正確的標記，且範圍要超過 50% (當然愈高愈好)

在裝置上更新 ML Model 的方法

並不需要每次 model 有更新，就要進版，也可以使用網路上下載的方式進行 model 的更新。

```
let compiledUrl = try MLModel.compileModel(at: modelUrl)
let model = try MLModel(contentsOf: compiledUrl)
```

Compile 完成後，可以使用 FileManager 把 model 存在裝置上。

[Apple Documentation](#)



關鍵字

你如果對原理有興趣
你可以找左邊關鍵字的社群

Tensor Flow

CNN

類神經網路

NLP

Model-based

蒙地卡羅算法

Reinforcement Learning

Model Free

Policy

Q learning

Awesome CoreML GitHub Repo

<https://github.com/likedan/Awesome-CoreML-Models>

<https://github.com/SwiftBrain/awesome-CoreML-models>

這兩個 Repository 上有許多已經訓練好的 Core ML models

你可以針對特定用途，下載所需要的 Model

e.g. 年齡識別、目標識別、句子主題等等

<https://github.com/MoonAndEye/CoreMLDemo>



Create ML



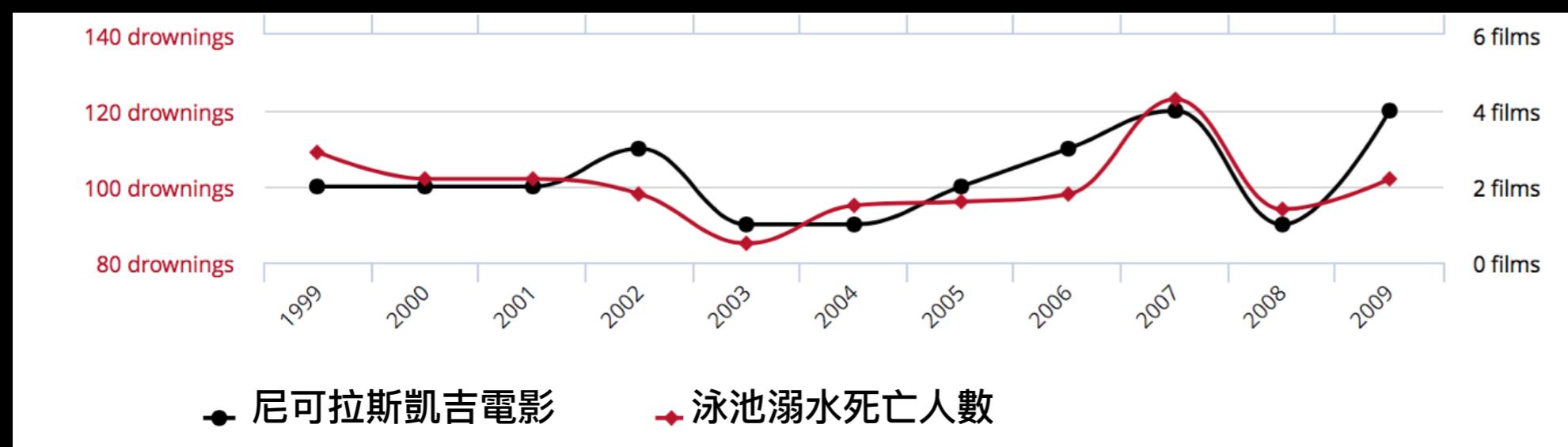
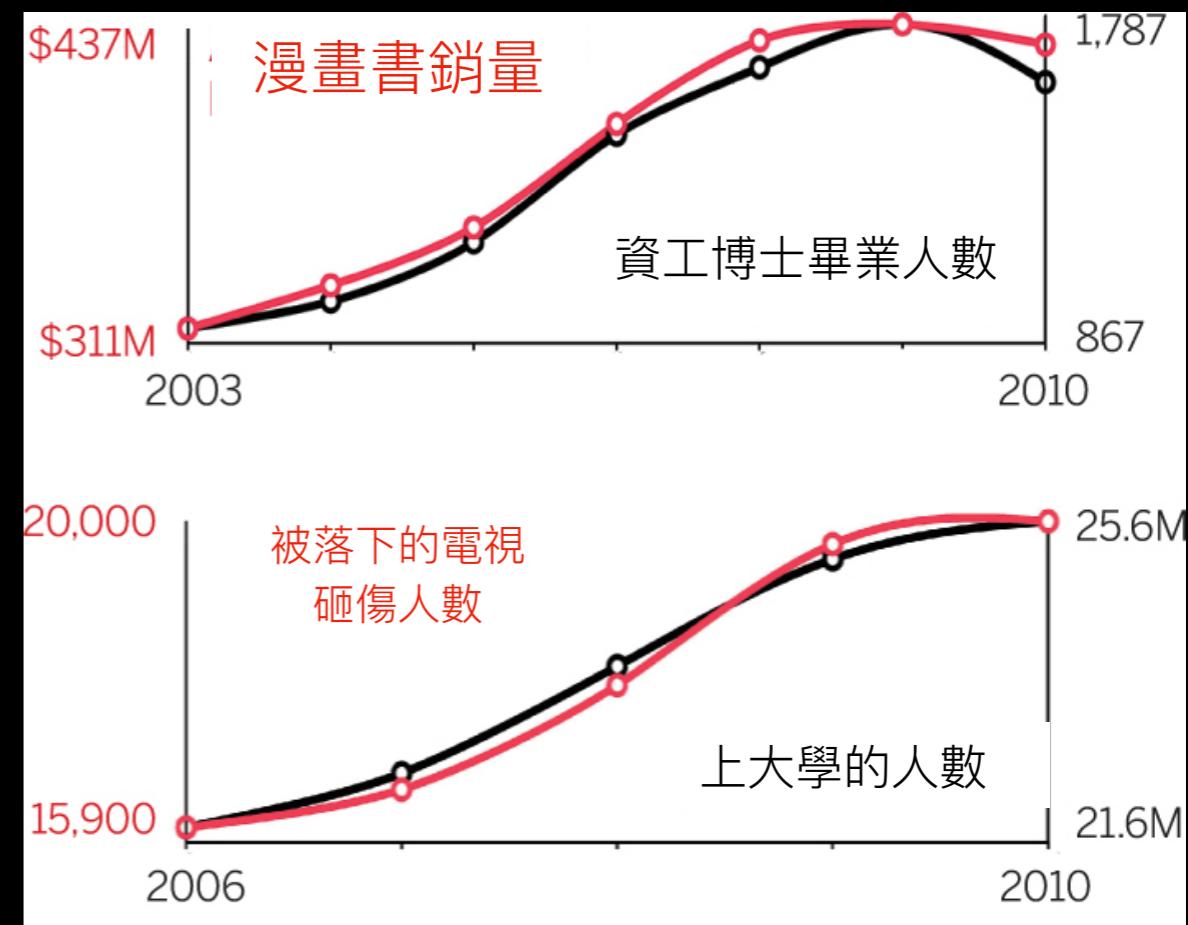
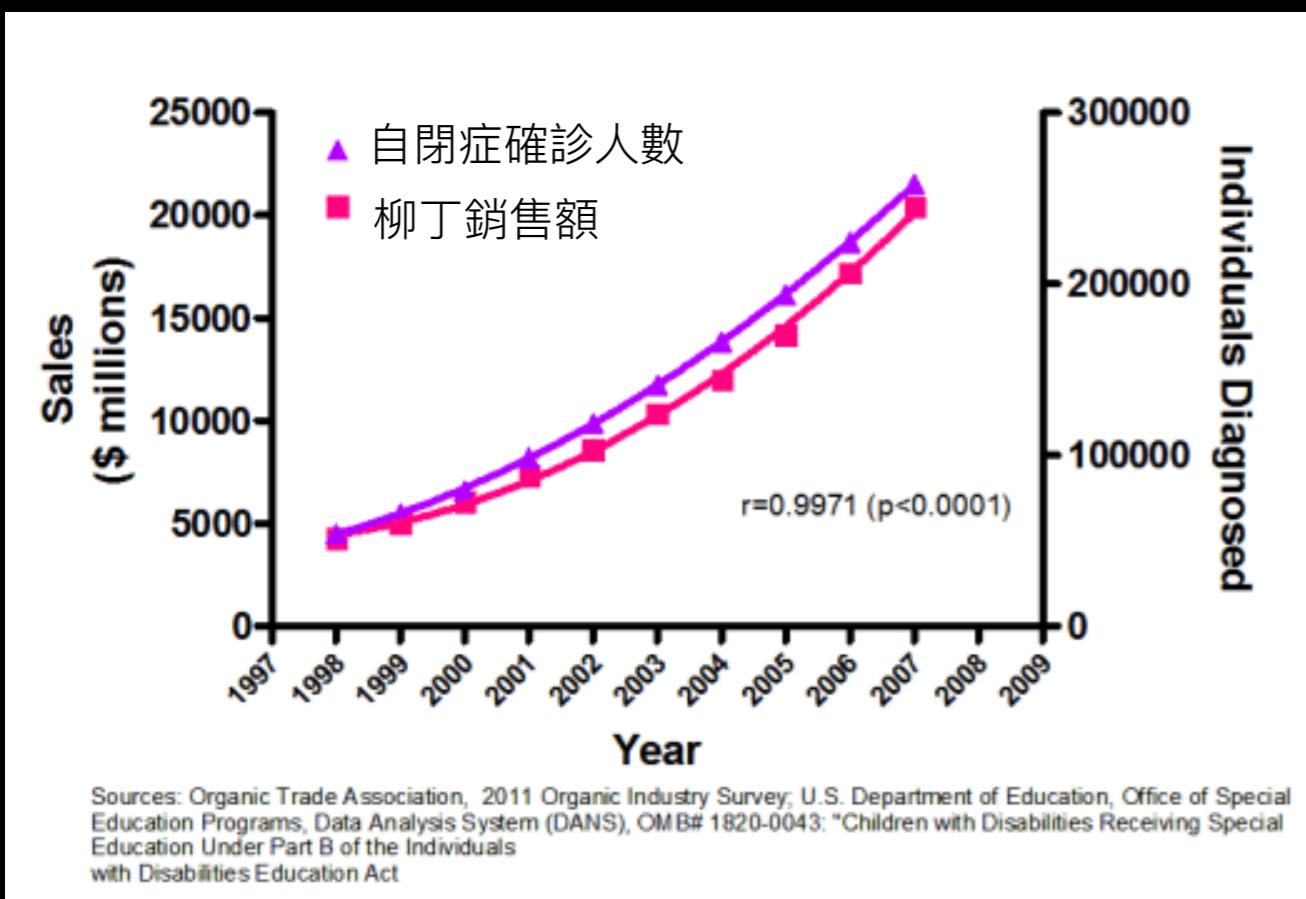
“Let’s build a ML model today”

<https://github.com/MoonAndEye/CoreMLDemo>

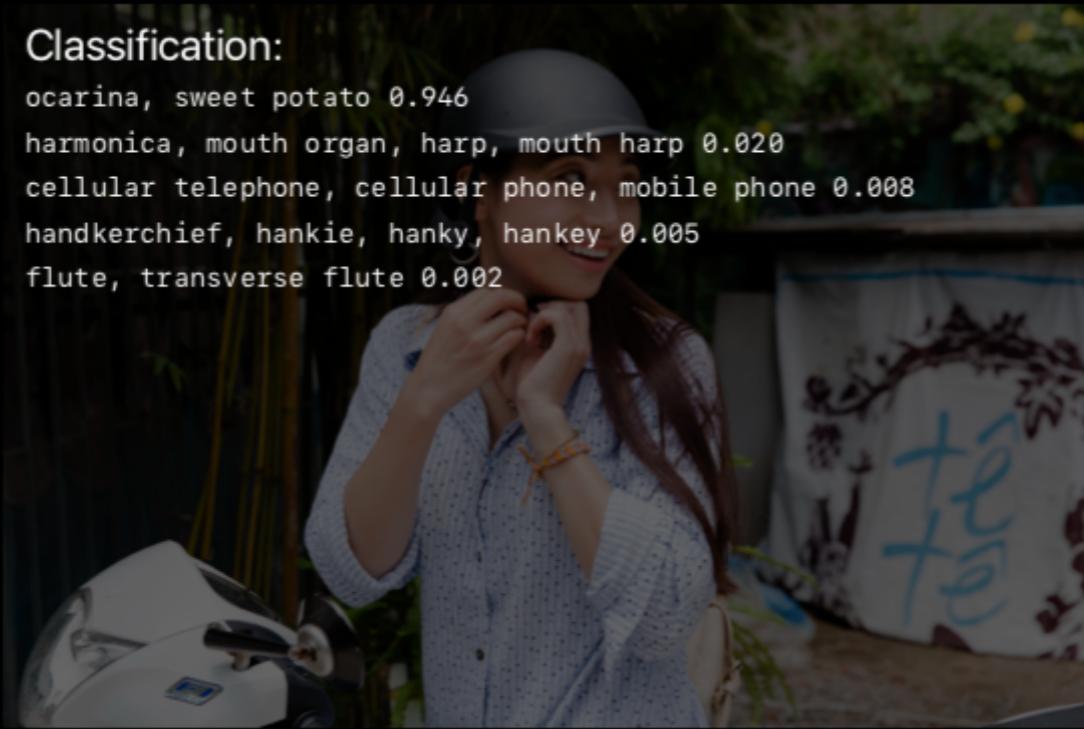
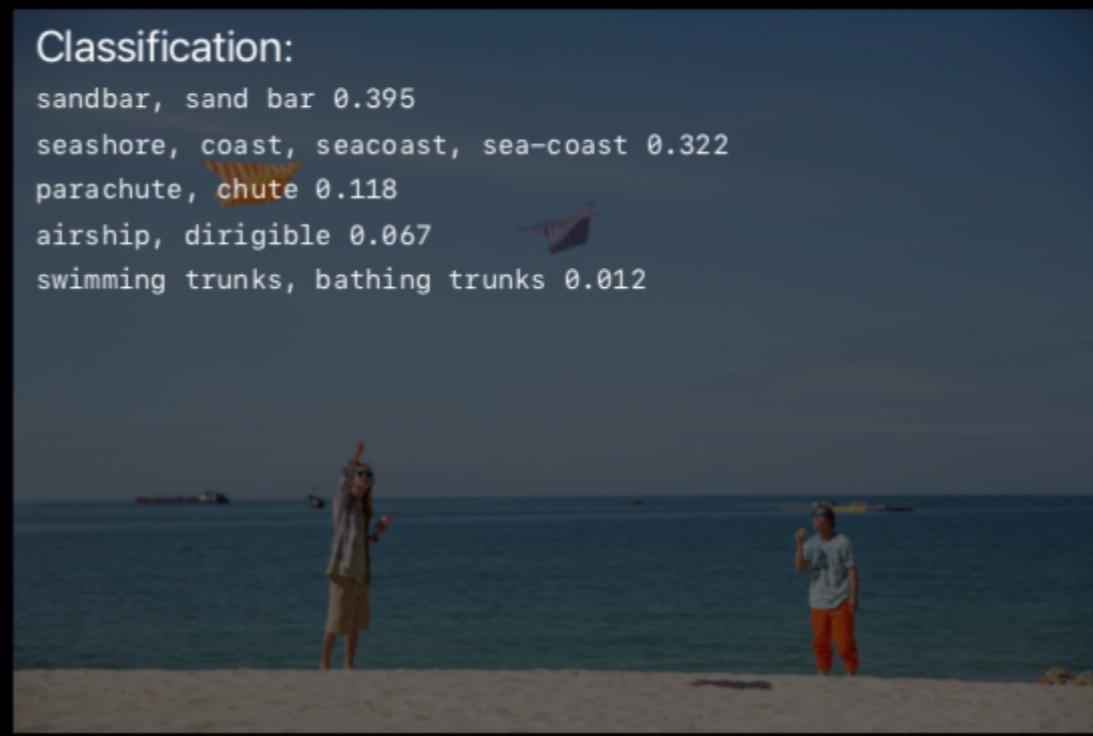
以下是沒辦法在時間內
放上去的 Slides

相關不為因果 (Correlation does not imply causation)

即使你算出來的數據上有相關，且相關係數高達 0.9 以上，也不能代表這兩組數據有關聯。



信心 (Confidence Score) 代表什麼?



```
// Vision/ObjectRecognitionViewController.swift
// Created by Frank Dzepka on 5/26/17.
// https://github.com/moonandeye/CoreMLDemo

import Vision

class VisionObjectRecognitionViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        self.title = "Vision Object Recognition"
    }

    @IBAction func setupVision(_ sender: Any) {
        self.setupVision()
    }

    func setupVision() {
        let error: NSError? = nil

        guard let modelURL = Bundle.main.url(forResource: "ObjectDetector", withExtension: "mlmodel") else {
            return NSError(domain: "VisionObjectRecognitionViewController", code: -1, userInfo: [NSLocalizedDescriptionKey: "Model file is missing"])
        }

        do {
            let visionModel = try VNCoreMLModel(for: MLModel(contentsOf: modelURL))
            let objectRecognition = VNCoreMLRequest(model: visionModel, completionHandler: { (request, error) in
                DispatchQueue.main.async(execute: {
                    // perform all the UI updates on the main queue
                    if let results = request.results {
                        self.drawVisionRequestResults(results)
                    }
                })
            })
            self.requests = [objectRecognition]
        } catch let error as NSError {
            print("Model loading went wrong: \(error)")
        }
    }

    return error
}
```

NLP Classifier

**The energy of developers
is amazing!**



Sentiment Analysis

情感分析

**Deposed Prince wants to
give you money
SPAM**

Spam Analysis

垃圾郵件分析

**The Warriors just had an
amazing comeback win
SPORT**

Topic Analysis

句子主題分析

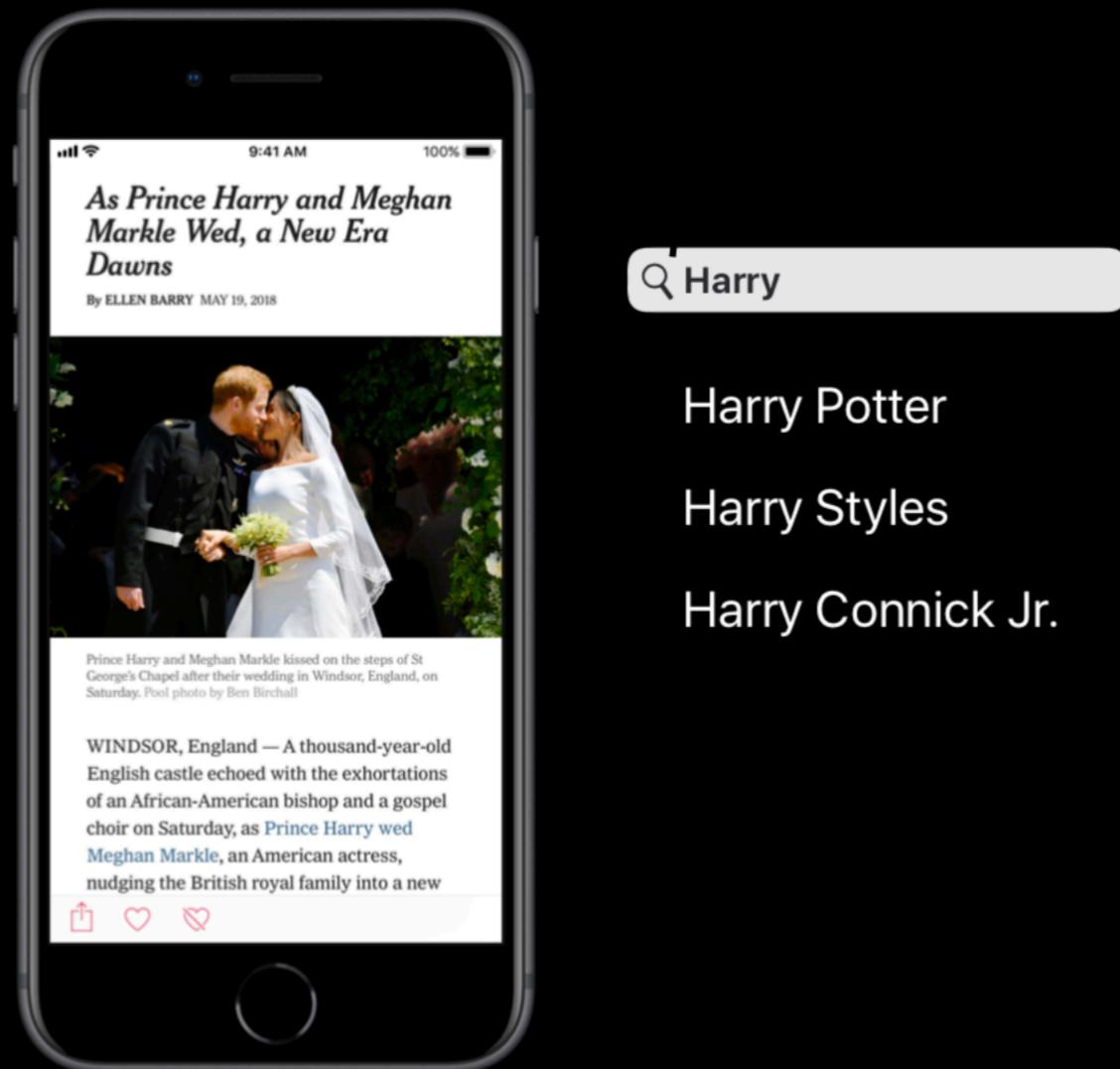
分析你句子中的主題

CSV

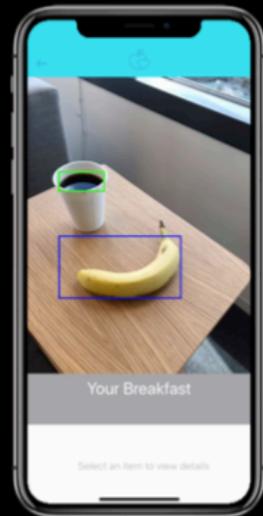
QueriesWithDomainLabels.csv

I'm looking for a place to stay in Barcelona., HOTELS
Where can I get good Mexican food on a Sunday?, RESTAURANTS
Find me an inexpensive round trip flight to London., FLIGHTS

命名識別 - 情境



What Do These Apps Have in Common?



偵測並辨視物體



即時偵測



手勢操作



推薦個性化鬍子



風格變換

使用 Machine Learning

只需要少量的 Data 就可以訓練了

從 Turi Create 做出來的 Model 可以在 CoreML 中使用

五個步驟就可以做出 Model (低於 20 行程式碼, 4 行輸出 SFrame, 7 行輸出 model)

命名識別

Named Entity Recognition

```
import NaturalLanguage

let tagger = NLTagger(tagSchemes: [.nameType])

let str = "Prince Harry and Meghan Markle had their wedding ceremony in Windsor"

let strRange = str.startIndex ..< str.endIndex

tagger.string = str

tagger.setLanguage(.english, range: strRange)

let tags = tagger.tags(in: strRange, unit: .word,
                      scheme: .nameType, options: .omitWhitespace)
```

PER: Prince Harry
Meghan Markle

LOC: Windsor

Positive Post App

- WWDC18 上 Demo 的 App

<https://github.com/MoonAndEye/CoreMLDemo>

表格型資料預測

- 依照輸入的項目，求出迴歸曲線。

房價預測

No. of bedrooms
No. of baths
Location
Sq. ft
Lot size

紅酒風味預測

Acidity
Sugar
pH
Alcohol
Citric

喝酒地點預測

Rating
Price
Happy hour
Distance

- 依照輸入的項目，求出迴歸曲線。

波士頓房價中位數預測

7:03

每宅房間平均數	收入未達某標準人口佔比	師生比例
10.0	5.5	20

開始計算房價中位數

波士頓房價中位數

USD \$:690827

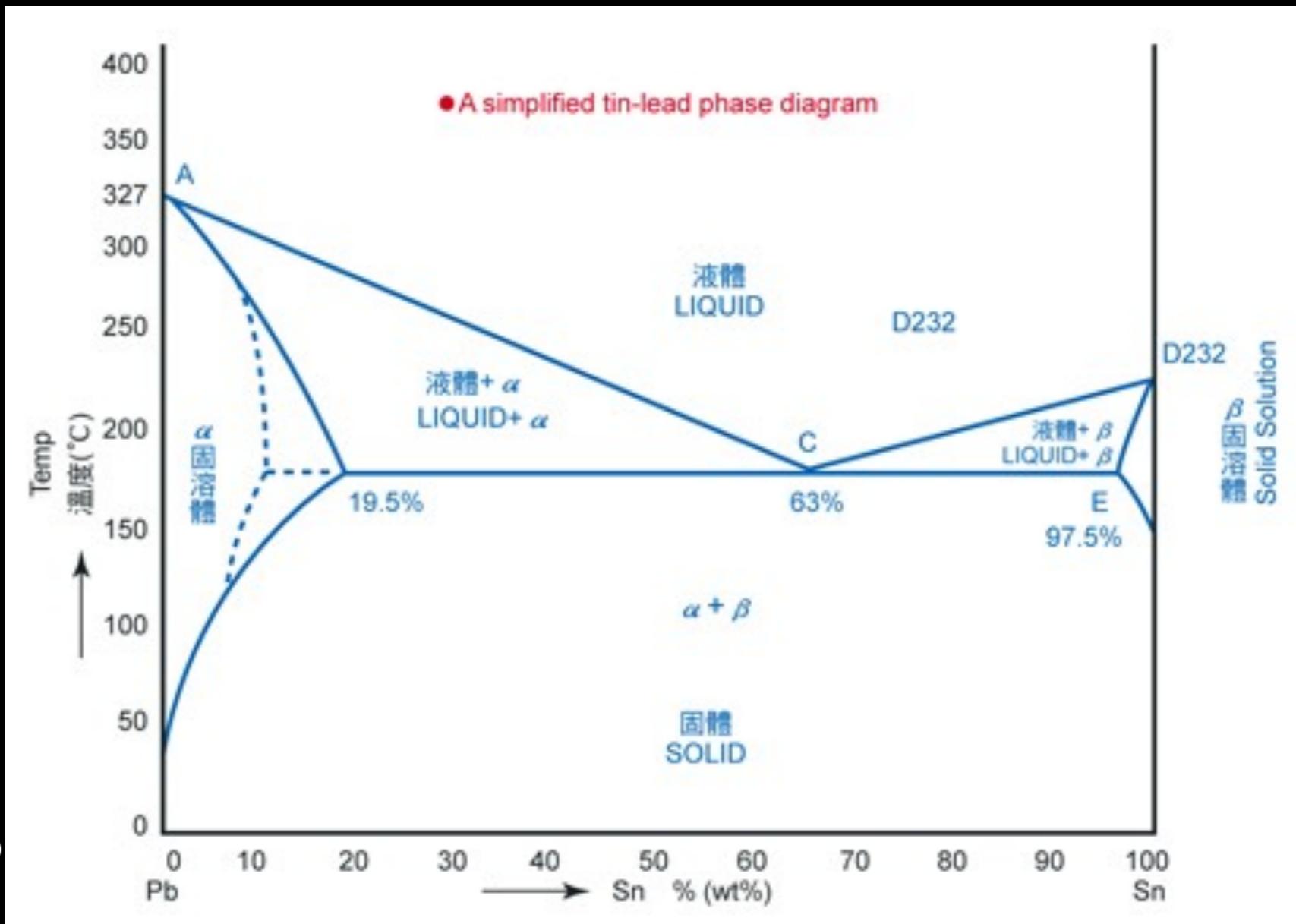
7. Attribute Information:

1. CRIM per capita crime rate by town	1. 城镇人均犯罪率
2. ZN proportion of residential land zoned for lots over 25,000 sq.ft.	2. 住宅用地所占比例, 25000英尺
3. INDUS proportion of non-retail business acres per town	3. 城镇中非商业用地的所占比例
4. CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)	4. CHAS查尔斯河虚拟变量, 用于回归分析
5. NOX nitric oxides concentration (parts per 10 million)	5. 环保指标
6. RM average number of rooms per dwelling	6. 每栋住宅的房间数
7. AGE proportion of owner-occupied units built prior to 1940	7. 1940年以前建成的自住单位的比例
8. DIS weighted distances to five Boston employment centres	8. 距离五个波士顿就业中心的加权距离
9. RAD index of accessibility to radial highways	9. 距离高速公路的便利指数
10. TAX full-value property-tax rate per \$10,000	10. 每一万美元的不动产税率
11. PTRATIO pupil-teacher ratio by town	11. 城镇中教师学生比例
12. B $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town	12. 城镇中黑人比例
13. LSTAT % lower status of the population	13. 地区有多少百分比的房东属于是低收入阶层
14. MEDV Median value of owner-occupied homes in \$1000's	14. 自住房屋房价的中位数

但，表格型資料不一定需要用到 AI 來分析

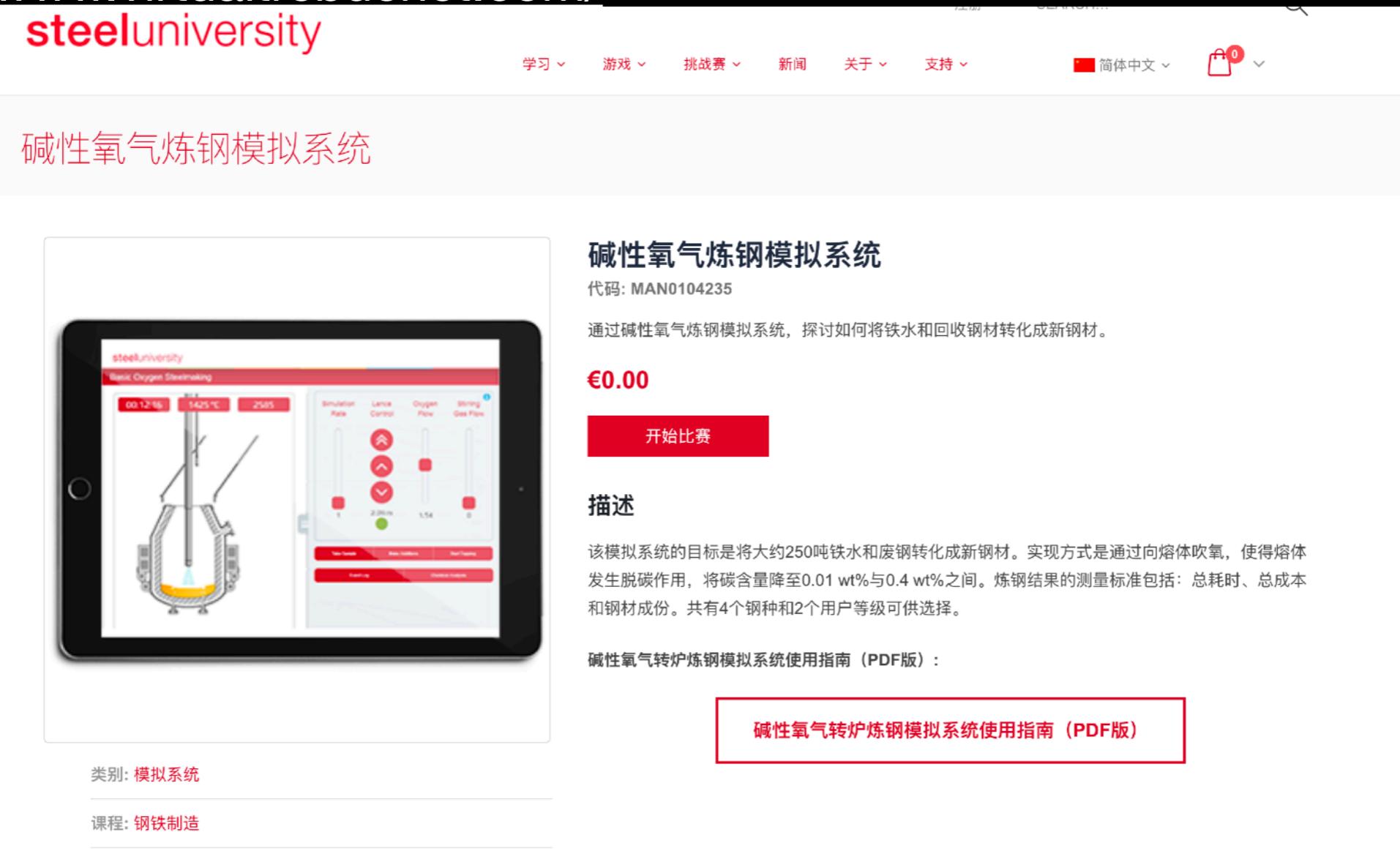
在使用前，可以先看看 D.O.E 或是田口法適不適用。

在使用前，可以先看看文獻上有沒有實驗數據可以用。(以鋅錫為例)



如果你手上沒有一條生產線 你可以用模擬器

- 世界鋼鐵大學
<https://steeluniversity.org>
- 模擬投石器 1
<http://www.virtualtrebuchet.com/>
- 模擬投石器 2
<https://www.geogebra.org/m/AHYnuZ8s>



The screenshot shows the steeluniversity website with the title "碱性氧气炼钢模拟系统" (Basic Oxygen Steelmaking Simulation) in red. Below it is a red button labeled "开始比赛" (Start Simulation). To the left, there's a mobile device displaying the simulation interface, which includes a diagram of a blast furnace and various control sliders and inputs. At the bottom, there are category labels: "类别: 模拟系统" (Category: Simulation System) and "课程: 钢铁制造" (Course: Steel Manufacturing).

steeluniversity

学习 游戏 挑战赛 新闻 关于 支持 简体中文 0

碱性氧气炼钢模拟系统

代码: MAN0104235

通过碱性氧气炼钢模拟系统，探讨如何将铁水和回收钢材转化成新钢材。

€0.00

开始比赛

描述

该模拟系统的目标是将大约250吨铁水和废钢转化成新钢材。实现方式是通过向熔体吹氧，使得熔体发生脱碳作用，将碳含量降至0.01 wt%与0.4 wt%之间。炼钢结果的测量标准包括：总耗时、总成本和钢材成份。共有4个钢种和2个用户等级可供选择。

碱性氧气转炉炼钢模拟系统使用指南（PDF版）:

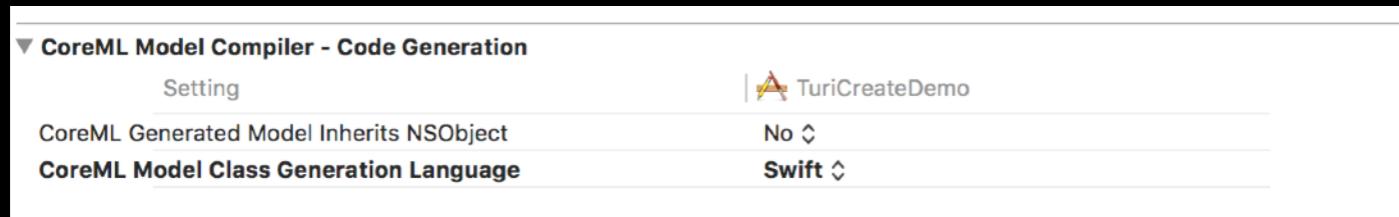
碱性氧气转炉炼钢模拟系统使用指南（PDF版）

https://g

一些你可能會踩到的雷

- 如果你裡面有兩個以上的 Model, 一開始可能會 build 不起來

Build setting 把 Generation Language 調成 Swift (預設 Automatic)



- 有時候 Model 在 build 生出來的 swift 檔裡面會有類別未宣告的 error

清掉 DerivedData 再 build 一次

- 使用 style transfer 的時候, max_iteration 預設是 5000 (有可能跑到兩天, 這台 MBP 設定 1000 的時候大約會花上 6 小時)

A GitHub comment from user [enzyme69](#) on June 15, 2015. The comment discusses the performance of style transfer with 5000 iterations:

A bit of note for anyone trying to run on CPU 5000 iterations:

- Perhaps 5000 is not always the default
- 5000 took me around 2 days
- 1000-1600 iterations is probably enough. 5000 is so much cleaner less noise, but overall look is similar.
- 100 iterations will give just a standard filter like.

I am not achieving mosaic look, however, getting some really nice comic and watercolor look, when using Gustav Klimt artworks as Style.