

# Core ML 相關 Sessions

## 濃縮呈現

Marvin Lin

# 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).



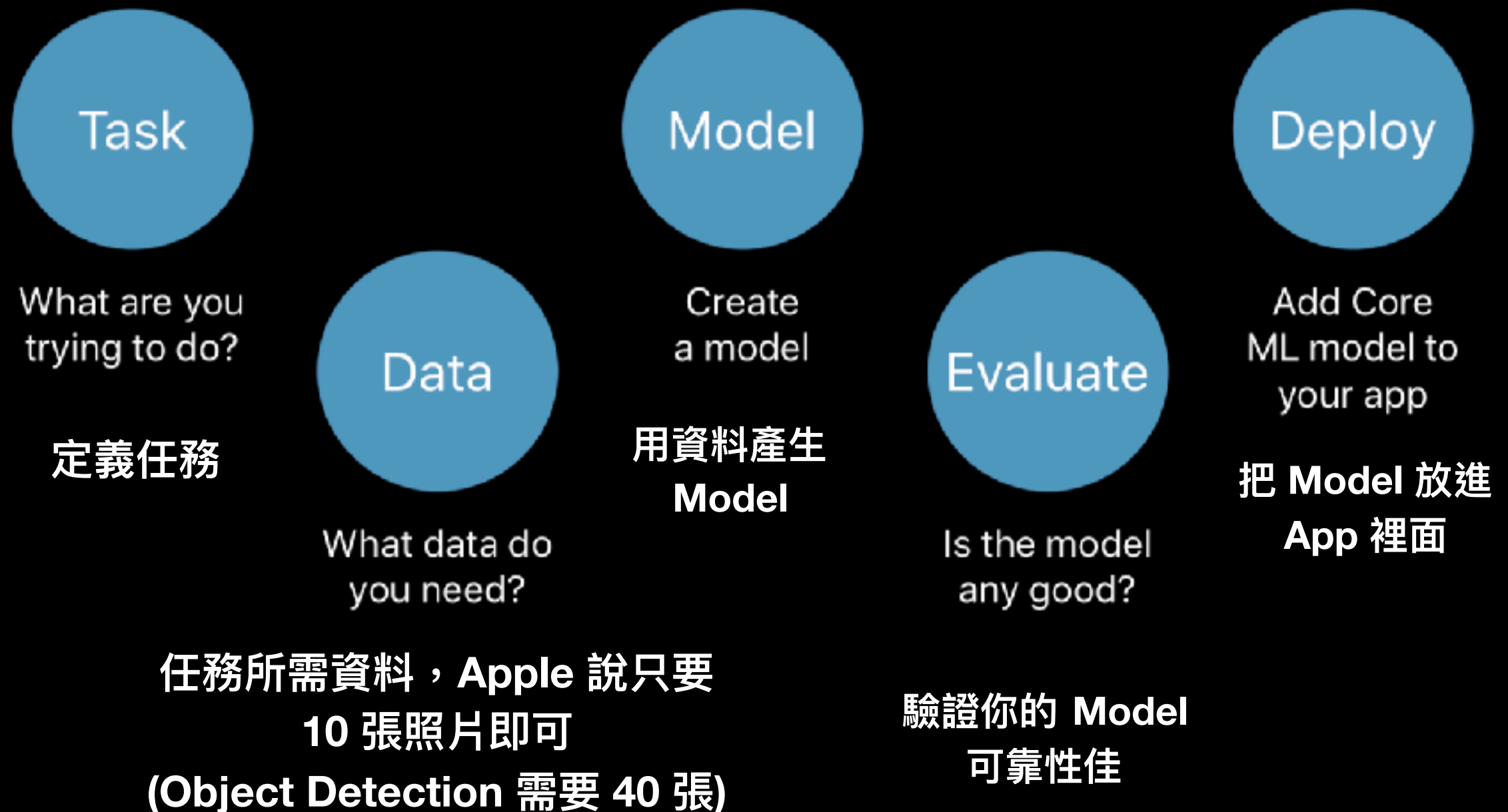
## Session 716 & 717 - Vision (部分內容)

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?



Simple model creation

Tailored to your app

Leverages core Apple technologies

Powered by Mac

圖案分類



Label

輸出分類類別

文章分類



Label

輸出分類類別

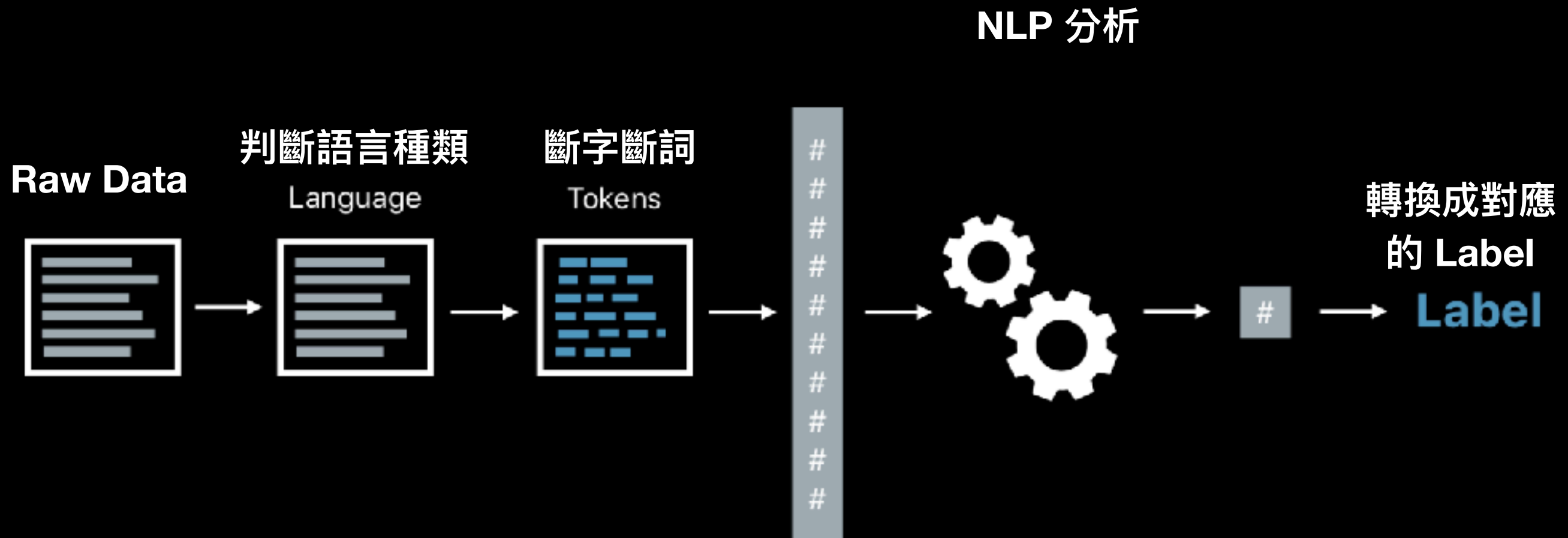
(結構化資料) 表格型數據



Number

輸出數字結果

# 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

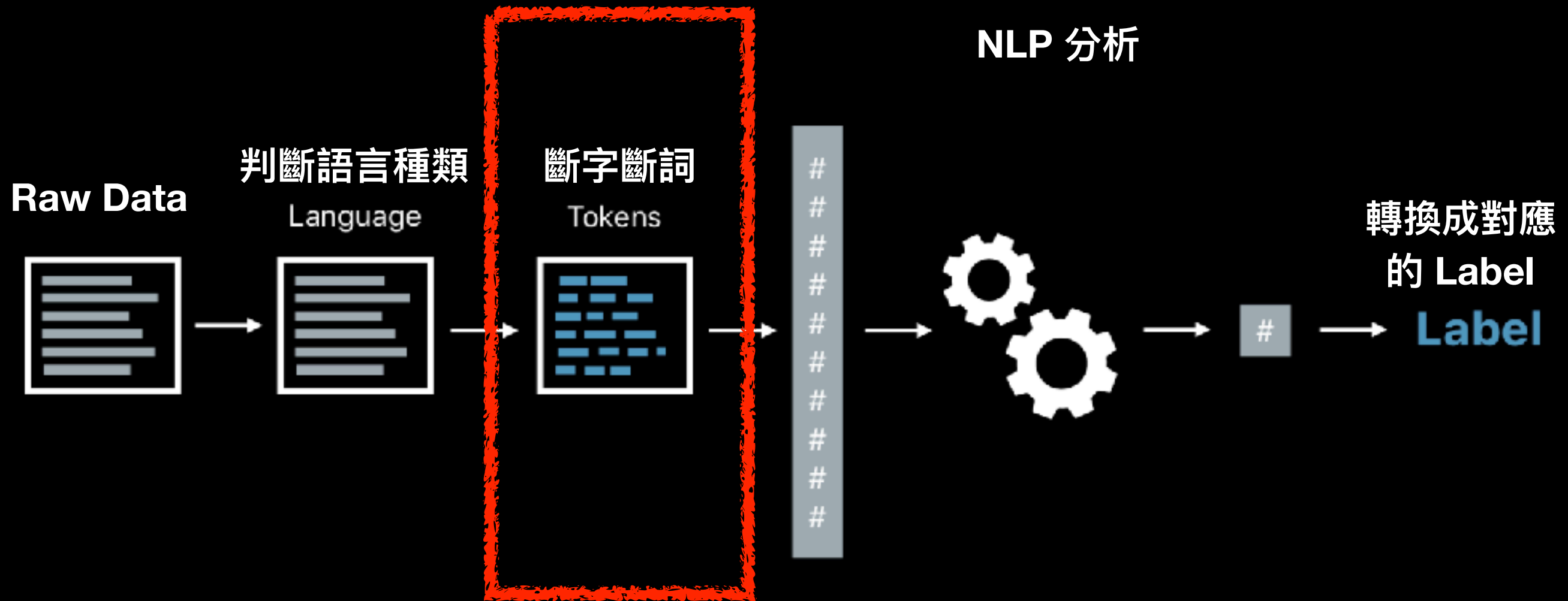
歐洲語系沒有問題，但是馬來語和印尼語無法正確判斷

```
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(
```

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


```
44 let tokenizer
45 let str = "き
46 let strRange
47 tokenizer.st
48 // tokens 會
```

きょう  
の  
ぼく  
に  
げんかい  
は  
ない


```
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
```

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

*dmlc*  
**XGBoost**

 Keras

Caffe

 *scikit*  
*learn*

**LIBSVM**

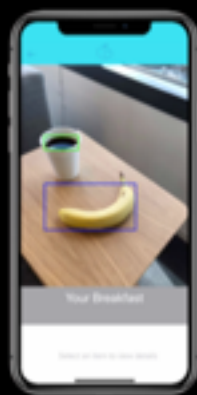
 mxnet

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

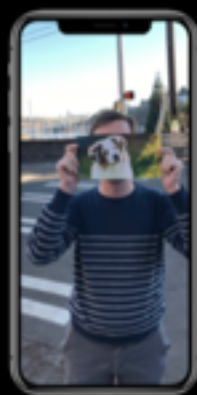
# Turi Create 是什麼？

- 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>



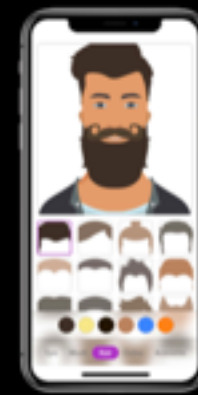
偵測並辨視物體



即時偵測



手勢操作



推薦個性化鬍子



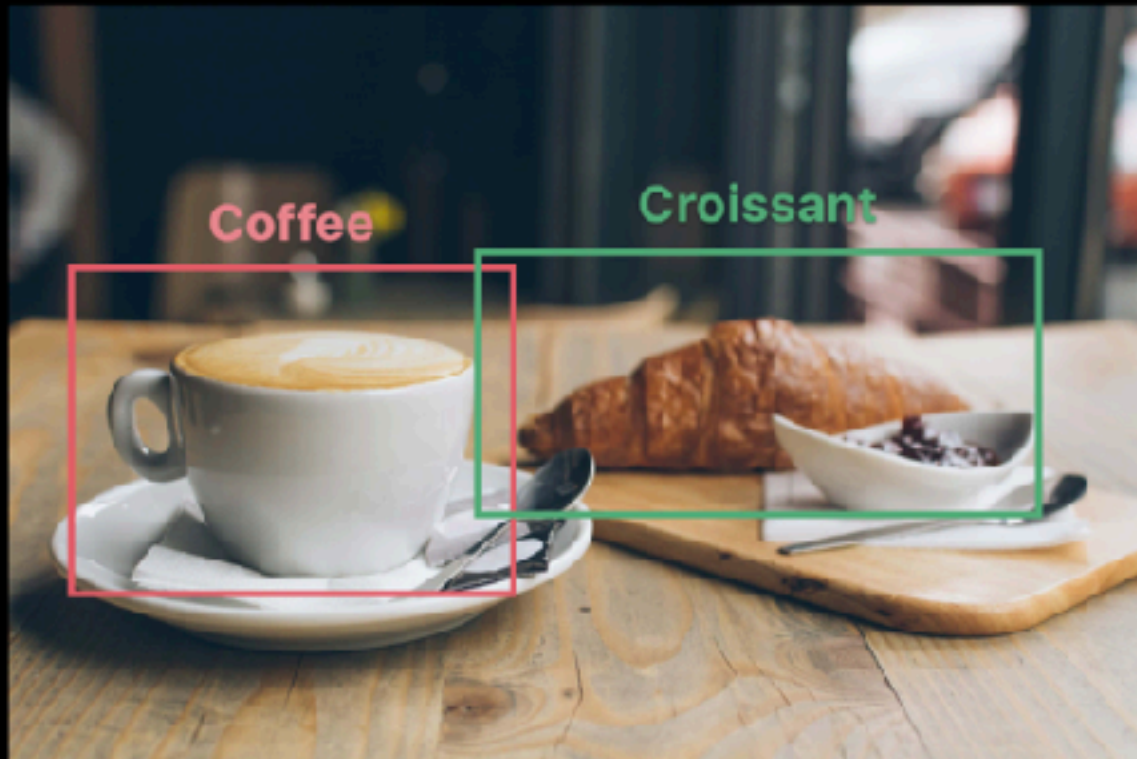
風格變換

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

# Jupyter Notebook

- Image Classifier
- Style Transfer
- Object Detection

# Object Detection



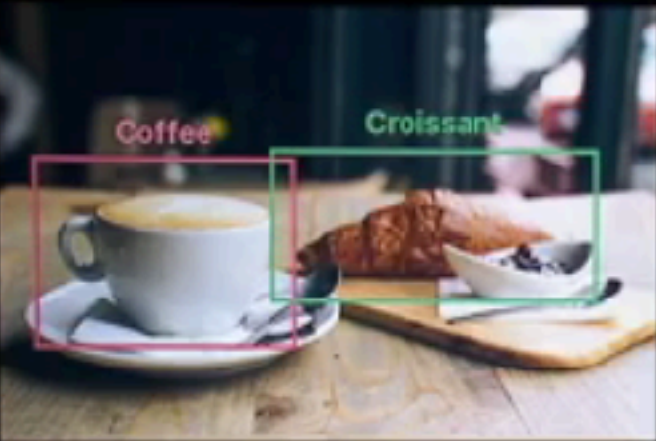
```
[
  {
    "label"      : "coffee",
    "coordinates" : {
      "x"      : 387,
      "y"      : 660,
      "height"  : 550,
      "width"   : 814,
    }
  },
  {
    "label"      : "croissant",
    "coordinates" : {
      "x"      : 800,
      "y"      : 630,
      "height"  : 373,
      "width"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

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

# 你的 X, Y 是我的 X, Y 嗎？



The slide displays a photograph of a coffee cup and a croissant on a wooden table. The coffee cup is enclosed in a red bounding box, and the croissant is enclosed in a green bounding box. Above the coffee cup, the word "Coffee" is written in red, and above the croissant, the word "Croissant" is written in green. To the right of the image, there is a JSON object representing the bounding box coordinates for the coffee cup:

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

Where x and y refer to the

center of that bounding box.

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

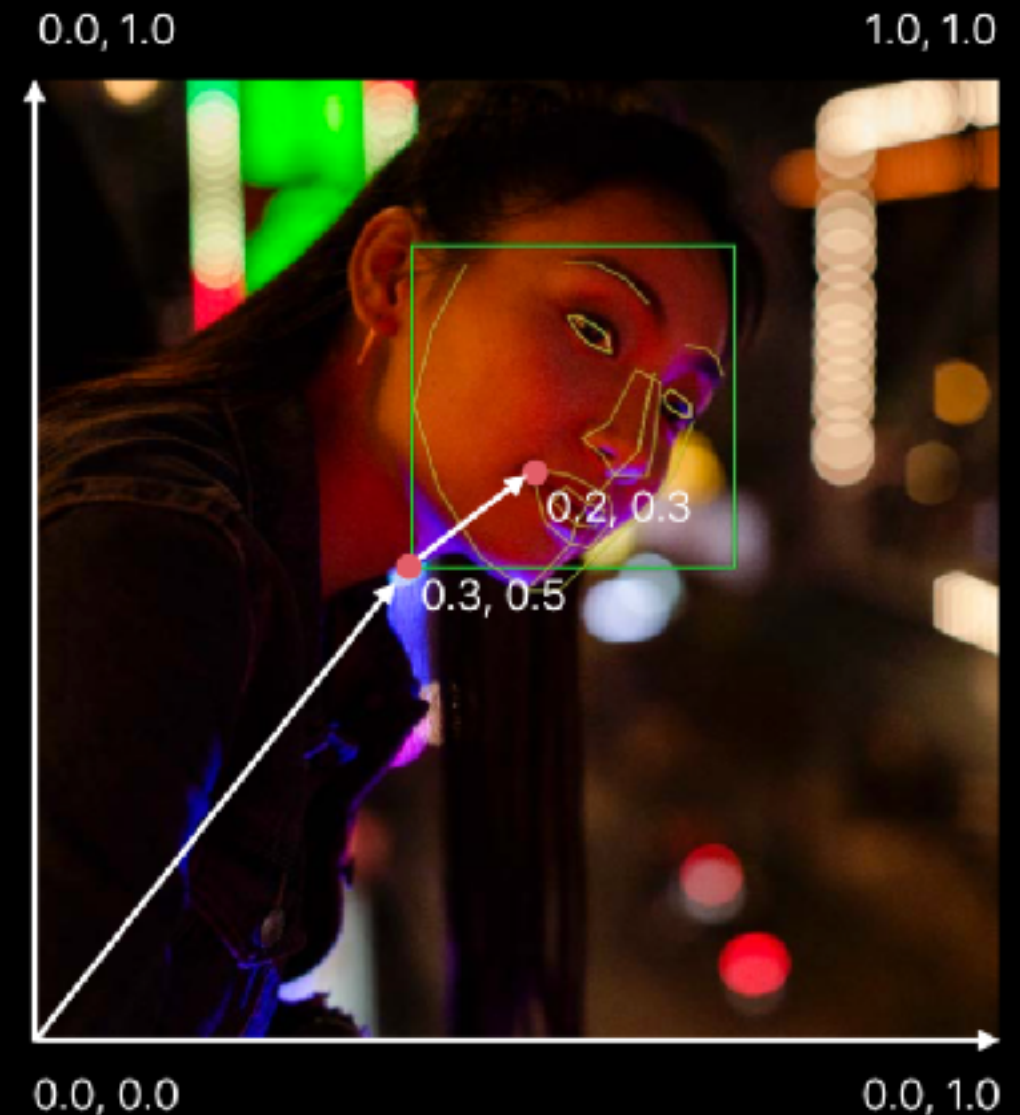
原點在左下角

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**

**Q learning**

**Policy**

# 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



Image  
Classification

Object  
Detection

Recommend

Activity  
Classification

Image  
Similarity

Style  
Transfer

Text  
Classifier

Text  
Topics

Classifiers

Regression

Clustering

Similarity

**“Let’s build a ML model today”**

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