

課程

深度學習實作與應用

授課老師

黃意婷

# 愛文芒果—— 利用影像進行等級分類

Group 2

R10725003 黃柏叡 R10725028 黃啟宏 R10725050 潘躍升

# 問題描述

- 愛文芒果影像辨識正式賽的等級分類競賽
- 愛文芒果採收後依品質篩選
- A、B、C 三等級，依序為出口用、內銷用、加工用



# 問題描述

- 農村人口流失導致人力短缺
- 篩果流程因保鮮期壓縮地極短

建立影像辨識演算法模型



對愛文芒果影像進行三種等級分類

# Outline

- 資料描述
- 資料處理
- 建立模型
- 結果
- 未來展望

# 資料描述

- 資料集包含
  - 訓練資料集(Training Data and Validation Data) 5600筆
  - 測試資料集(Testing Data) 800筆
- 這兩資料集均包含
  - 影像圖檔(Image)
  - 各影像對應之等級標籤(Grade)

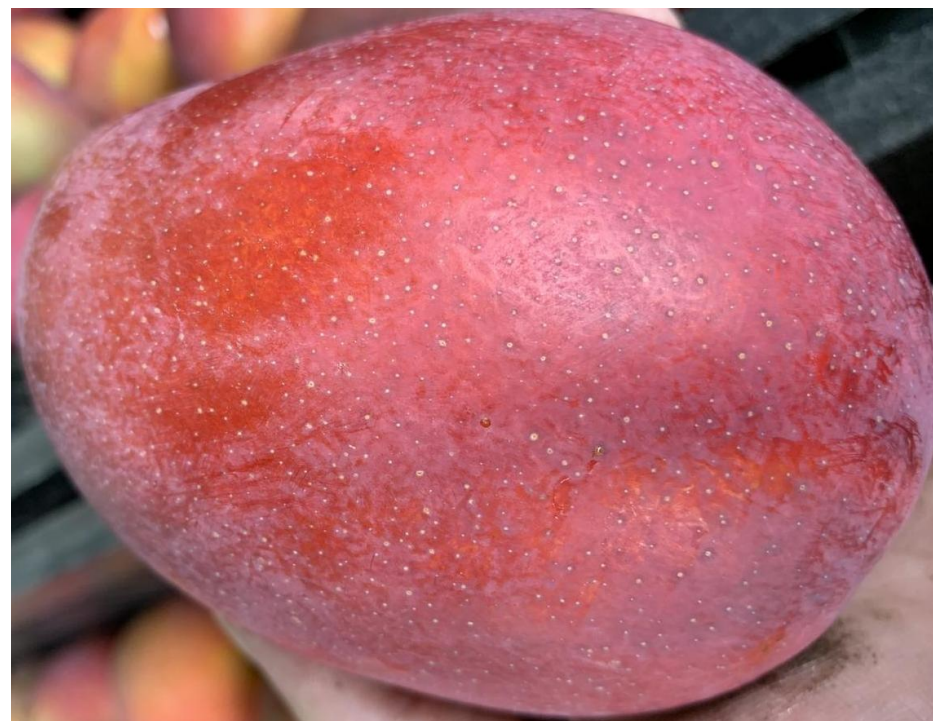


← train.csv		
	A	B
1	image_id	label
2	00002.jpg	C
3	00003.jpg	C
4	00004.jpg	B
5	00005.jpg	A
6	00007.jpg	A
7	00008.jpg	B
8	00009.jpg	B
9	00010.jpg	A

# 資料描述

- A: 出口用

- 色澤飽滿，外觀完整無損害痕跡





# 資料描述

- **B: 內銷用**

- 色澤較出口用黯淡，外觀黑點較明顯但瑕不掩瑜

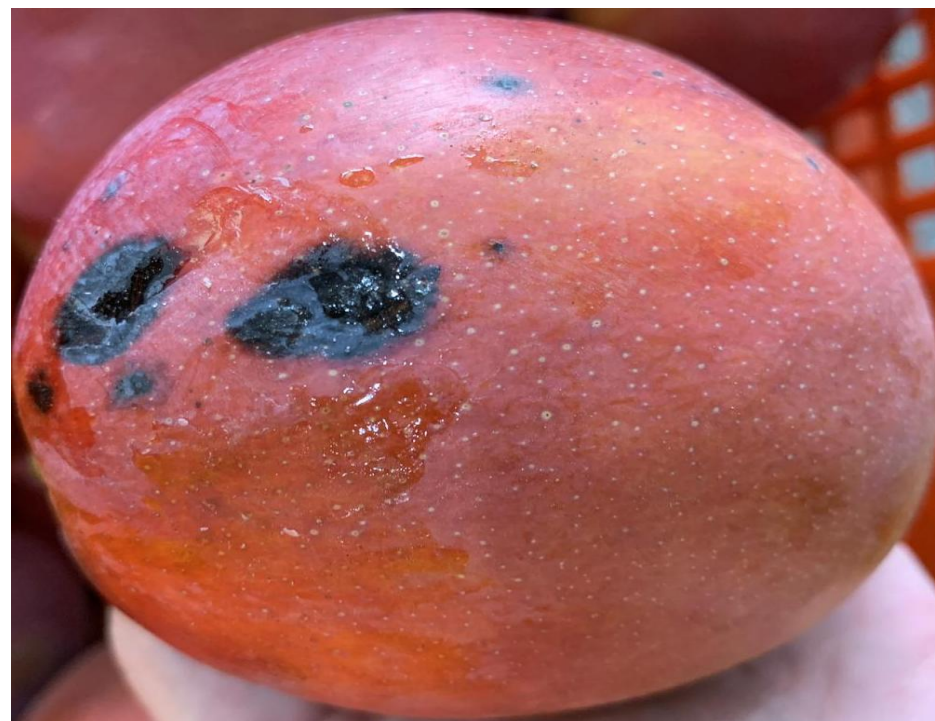




# 資料描述

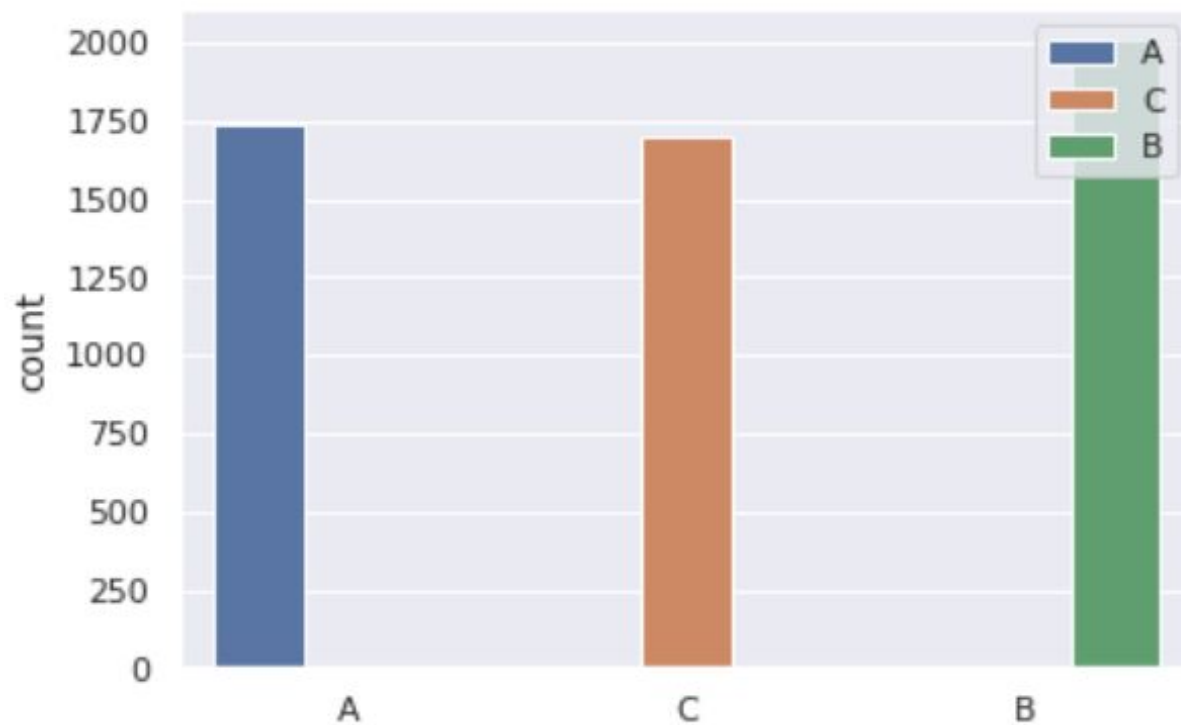
- C: 加工用

- 因黑斑病及炭疽病較嚴重導致賣相不好, 故多以加工使用



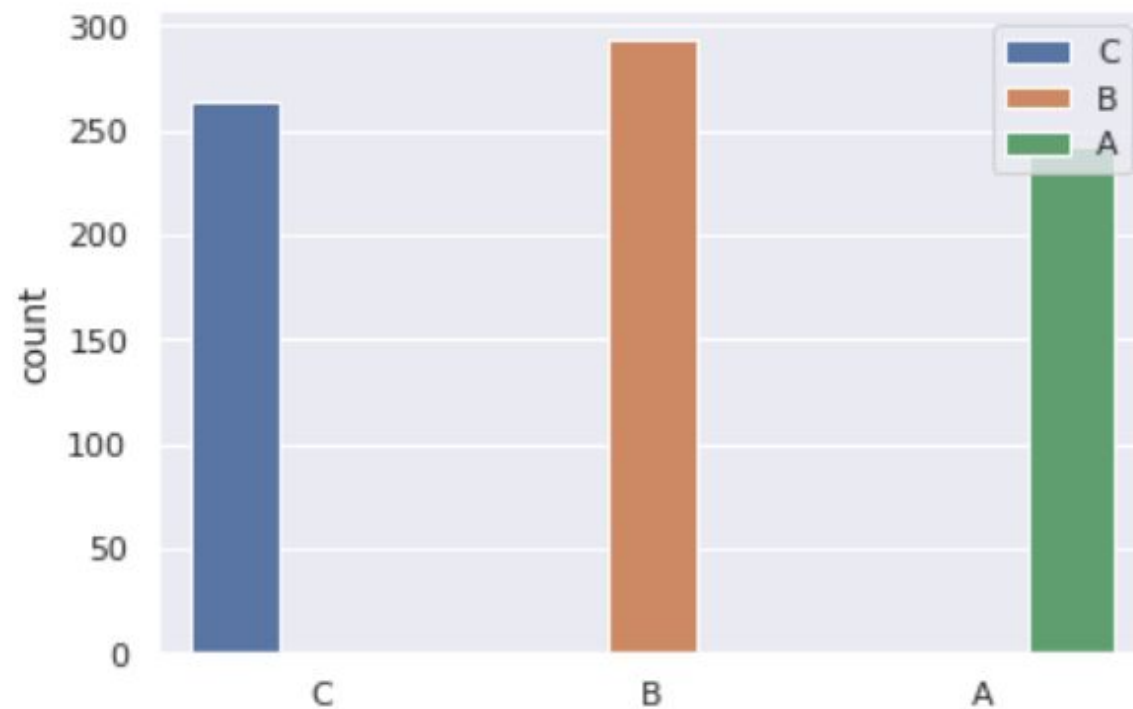
# 資料描述

- 訓練資料集(Training & Validation Data) 5600筆
  - 資料集分佈



# 資料描述

- 測試資料集(Testing Data) 800筆
  - 資料集分佈



# Outline

- 資料描述
- **資料處理**
- 建立模型
- 結果
- 未來展望

# 資料處理

## 目的為何？

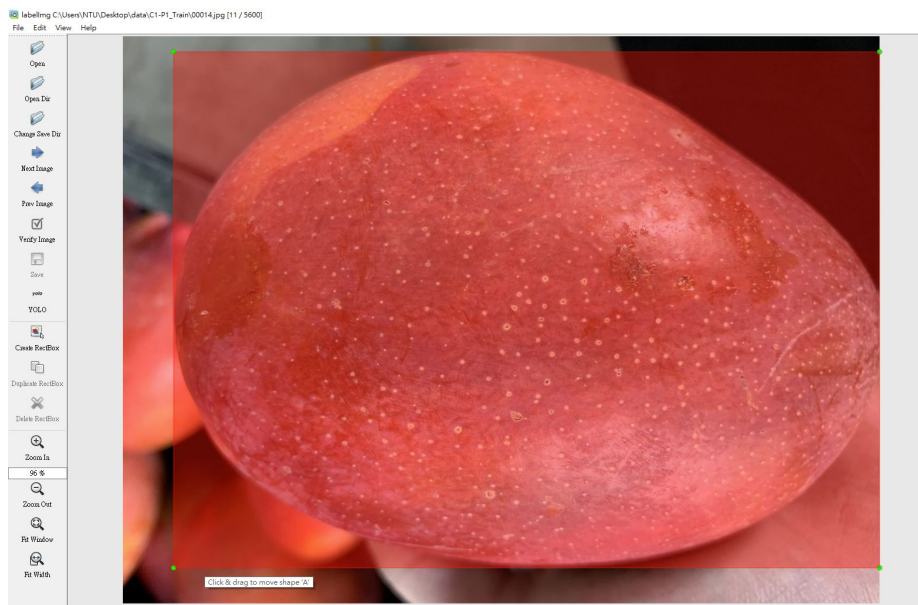
- 模擬果園真實情形
- 包含複雜背景



# 資料處理

- YOLOv5

- 透過 labellmg 軟體與線上之芒果資料集共 1500 張圖片作為YOLOv5 模型之訓練資料集
- 將跑完成之 best.pt 權重檔套用至 YOLOv5 之 detect.py 中進行芒果偵測
- 利用主要偵測物體通常位於照片正中央的概念，捕捉座標涵蓋圖片中心之物體進行裁剪

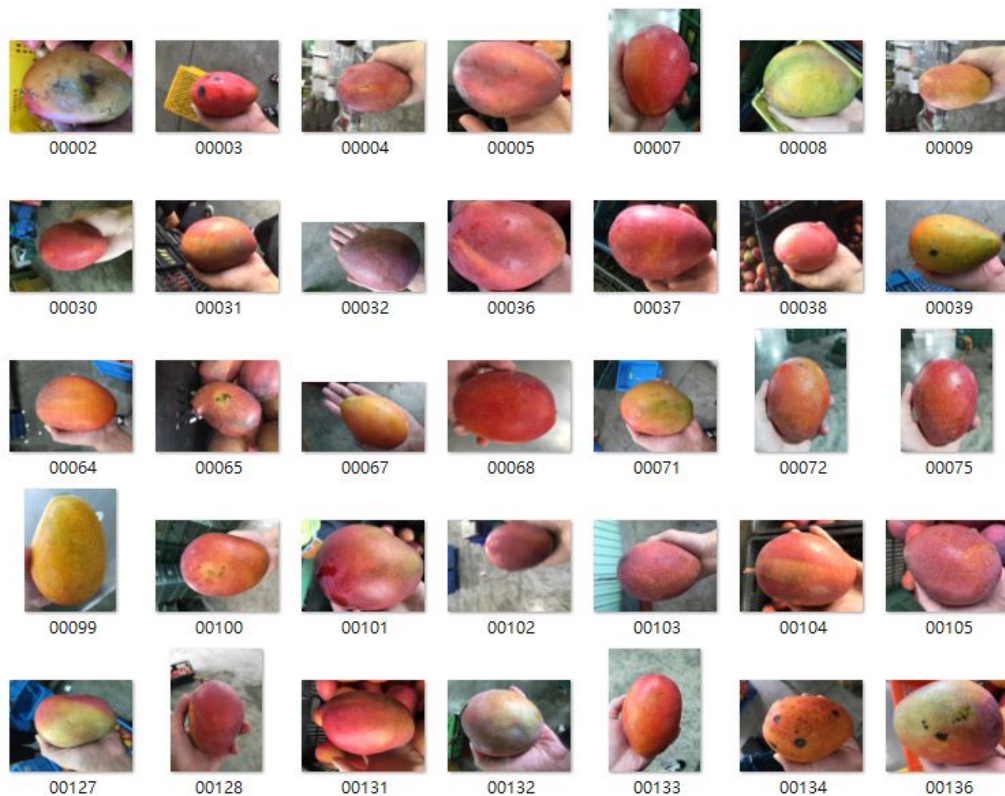


Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size			
5/19	0G	0.04788	0.02652	0	23	640: 100%		9/9	[00:48<00:00, 5
	Class	Images	Instances	P	R	mAP50	mAP50-95:	80%	277/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	80%	278/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	81%	279/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	81%	281/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	82%	283/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	82%	284/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	82%	285/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	83%	286/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	83%	287/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	83%	288/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	84%	289/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	84%	290/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	84%	291/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	84%	292/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	85%	293/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	85%	294/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	85%	295/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	86%	296/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	86%	298/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	87%	300/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	88%	304/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	88%	305/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	88%	306/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	89%	309/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	90%	310/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	90%	311/346
	Class	Images	Instances	P	R	mAP50	mAP50-95:	90%	312/346

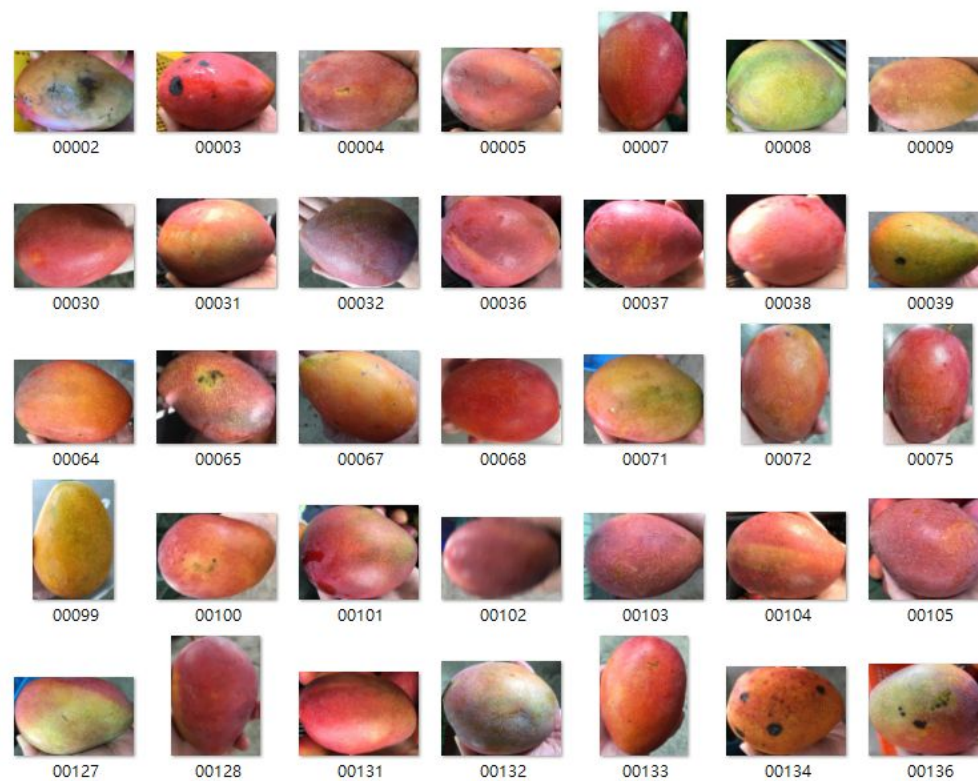


# 資料處理

- YOLOv5



before YOLO



after YOLO



# 資料處理

- 資料壓縮
  - $224 \times 224$





# Outline

- 資料描述
- 資料處理
- **建立模型**
- 結果
- 未來展望

# 建立模型

## CNN 模型

- 資料壓縮
- $N*3*224*224$
- Epoch 100 (early stop at 32)
- Accuracy 75.13%



Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 222, 222]	896
BatchNorm2d-2	[-1, 32, 222, 222]	64
ReLU-3	[-1, 32, 222, 222]	0
Conv2d-4	[-1, 32, 220, 220]	9,248
BatchNorm2d-5	[-1, 32, 220, 220]	64
ReLU-6	[-1, 32, 220, 220]	0
MaxPool2d-7	[-1, 32, 110, 110]	0
Conv2d-8	[-1, 32, 108, 108]	9,248
BatchNorm2d-9	[-1, 32, 108, 108]	64
ReLU-10	[-1, 32, 108, 108]	0
MaxPool2d-11	[-1, 32, 54, 54]	0
Conv2d-12	[-1, 64, 52, 52]	18,496
BatchNorm2d-13	[-1, 64, 52, 52]	128
ReLU-14	[-1, 64, 52, 52]	0
MaxPool2d-15	[-1, 64, 26, 26]	0
Linear-16	[-1, 1024]	44,303,360
Dropout-17	[-1, 1024]	0
ReLU-18	[-1, 1024]	0
Linear-19	[-1, 256]	262,400
Dropout-20	[-1, 256]	0
ReLU-21	[-1, 256]	0
Linear-22	[-1, 128]	32,896
Dropout-23	[-1, 128]	0
ReLU-24	[-1, 128]	0
Linear-25	[-1, 64]	8,256
Dropout-26	[-1, 64]	0
ReLU-27	[-1, 64]	0
Linear-28	[-1, 3]	195

---

Total params: 44,645,315

Trainable params: 44,645,315

Non-trainable params: 0

---

Input size (MB): 0.57

Forward/backward pass size (MB): 88.08

Params size (MB): 170.31

Estimated Total Size (MB): 258.96

---

# 建立模型

## CNN 模型 with YOLO5

- 資料壓縮、物件偵測
- $N*3*224*224$
- Epoch 100 (early stop at 64)
- Accuracy 76.13%

# 建立模型

## VGG16

- 資料壓縮
- Pretrain with ImageNet
- Fine tuning overall model
- Epoch 100(early stop at 19)
- Accuracy 79.37%



Conv2d-1	[-1, 64, 224, 224]	1,792
ReLU-2	[-1, 64, 224, 224]	0
Conv2d-3	[-1, 64, 224, 224]	36,928
ReLU-4	[-1, 64, 224, 224]	0
MaxPool2d-5	[-1, 64, 112, 112]	0
Conv2d-6	[-1, 128, 112, 112]	73,856
ReLU-7	[-1, 128, 112, 112]	0
Conv2d-8	[-1, 128, 112, 112]	147,584
ReLU-9	[-1, 128, 112, 112]	0
MaxPool2d-10	[-1, 128, 56, 56]	0
Conv2d-11	[-1, 256, 56, 56]	295,168
ReLU-12	[-1, 256, 56, 56]	0
Conv2d-13	[-1, 256, 56, 56]	590,080
ReLU-14	[-1, 256, 56, 56]	0
Conv2d-15	[-1, 256, 56, 56]	590,080
ReLU-16	[-1, 256, 56, 56]	0
MaxPool2d-17	[-1, 256, 28, 28]	0
Conv2d-18	[-1, 512, 28, 28]	1,180,160
ReLU-19	[-1, 512, 28, 28]	0
Conv2d-20	[-1, 512, 28, 28]	2,359,808

## Top Model

ReLU-21	[-1, 512, 28, 28]	0
Conv2d-22	[-1, 512, 28, 28]	2,359,808
ReLU-23	[-1, 512, 28, 28]	0
MaxPool2d-24	[-1, 512, 14, 14]	0
Conv2d-25	[-1, 512, 14, 14]	2,359,808
ReLU-26	[-1, 512, 14, 14]	0
Conv2d-27	[-1, 512, 14, 14]	2,359,808
ReLU-28	[-1, 512, 14, 14]	0
Conv2d-29	[-1, 512, 14, 14]	2,359,808
ReLU-30	[-1, 512, 14, 14]	0
MaxPool2d-31	[-1, 512, 7, 7]	0
AdaptiveAvgPool2d-32	[-1, 512, 7, 7]	0
Linear-33	[-1, 512]	12,845,568
Dropout-34	[-1, 512]	0
Linear-35	[-1, 256]	131,328
Dropout-36	[-1, 256]	0
BatchNorm1d-37	[-1, 256]	512
Linear-38	[-1, 3]	771

# 建立模型

## VGG16 with YOLO

- 資料壓縮、物件偵測
- Pretrain with ImageNet
- Fine tuning overall model
- Epoch 100 (early stop at 22)
- Accuracy 76.37%

# Outline

- 資料描述
- 資料處理
- 建立模型
- **結果**
- 未來展望



# 結果

	CNN	CNN + YOLO	VGG16 	VGG16 + YOLO
Accuracy	75.13%	76.13%	79.37%	76.37%
F1 - A	0.7542	0.7724	0.7671	0.7985
F1 - B	0.6822	0.6861	0.7435	0.6759
F1 - C	0.8235	0.8399	0.8790	0.8291
F1	0.7533	0.7661	0.7962	0.7678

# 結果

## 模型方面

- CNN
  - 經調整已足夠處理本資料集的資料
- YOLOv5
  - 提升並不顯著
  - 仍具泛用性和實用性
- VGG16
  - 表現只比 CNN 好上些許
  - 芒果分級不需過於複雜的模型

# 結果

## 資料方面

- 不同等級芒果的 F1-score 表現相差許多
  - C 最佳, A 次之, B 最差
- 推測原因如下
  - 模型學習 C 等級圖片上黑點或瑕疵的區塊剛好吻合人工識別芒果等級的方式
  - B 等級身為 A 和 C 的過渡等級, 在人工識別時標準並不統一

# Outline

- 資料描述
- 資料處理
- 建立模型
- 結果
- 未來展望

# 未來展望

- 多訓練一點模型做 Ensemble Learning
- 套用其他網路架構

