

YOLOV1 PASCAL VOC (VISUAL OBJECT CLASSES) 資料集

DAST329-16

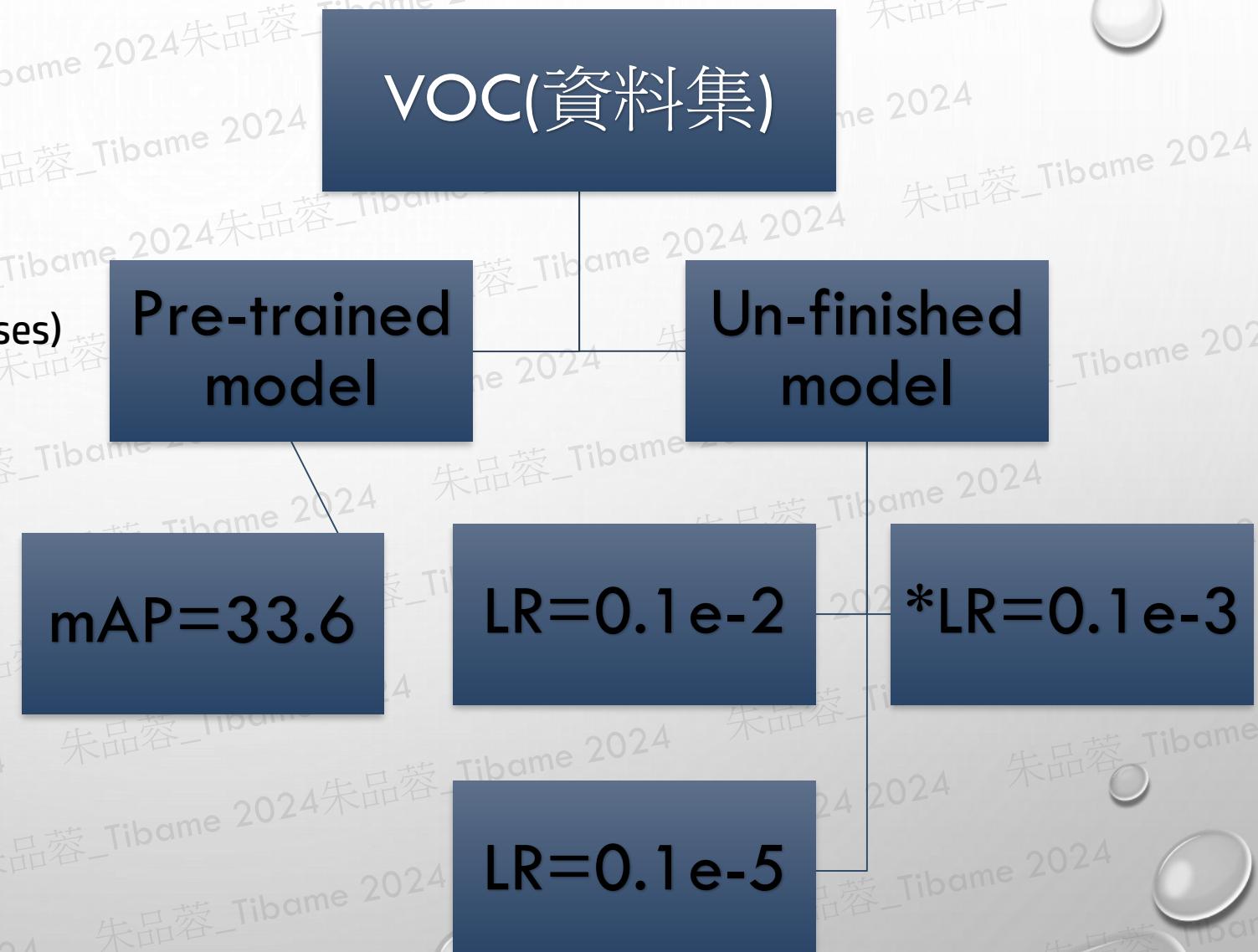
朱品蓉

實作8-1

BACKGROUND

VOC(資料集)

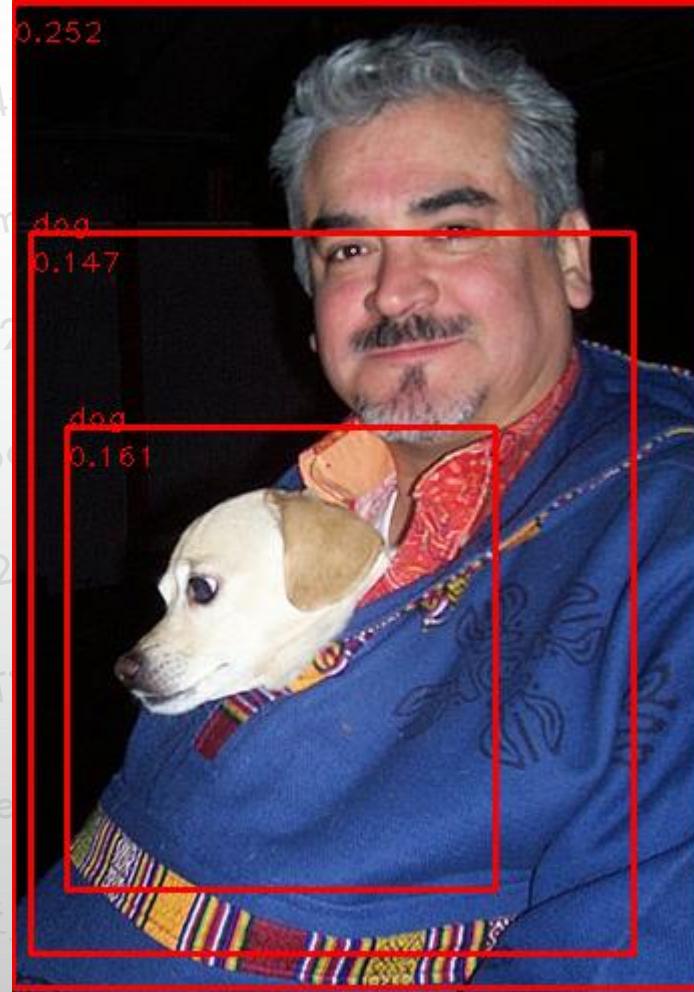
- 根據題目結構如圖
- 訓練模型的資料集為 PASCAL VOC (Visual Object Classes)
- pre-trained model
(YOLOv1+darknet19)
- Un-finished model
(with load_weight)/
(without load_weight)
需探討LR起始值對模型的影響



PRE-TRAIN MODEL (YOLOV1+DARKNET19)

根據所提供的最好條件下的模型:

- $Lr=0.1e-3$
- Epoch=150(程式碼中, 猜測)
- conf_thresh=0.1,nms_thresh=0.5
- mAP=33.6



OBSERVATION OF CODE

- 若要使用pre-trained darknet 19的weight

Epochs_start==1才能執行。(下圖)

- 但lr的起始值影響，要在epoch==0，才有效益，且後續的epoch，lr也有相對應的定義。(右圖)
- 故判斷為，在此程式碼下：

lr起始值對整體的mAP效益不大

```
# load pretrained model (Darknet-19) or unfinished trained model
if epochs_start == 1:
    net = YOLO(cls_num, bbox_num, box_scale, pretrained=pretrain_path)
else:
    net = YOLO(cls_num, bbox_num, box_scale)
```

```
#training code
for epoch in range(epochs_start, epochs_end):
    epoch_loss = 0
    train_iterator = tqdm(train_loader, ncols=30)
    mulit_batch_ = 0
    #print("epoch={}".format(epoch))
    if epoch == 0:
        learning_rate = lr/10      # lr = 0.1e-3
    elif epoch >= 1 and epoch <=5:
        learning_rate = 0.2e-3/10
    elif epoch <= 10:
        learning_rate = 0.2e-3/10
    elif epoch <= 30:
        learning_rate= 0.0001/10
    elif epoch <= 40:
        learning_rate=0.00001
    else:
        learning_rate=0.00001
```

DESIGN OF EXPERIMENT

在上述前提下，做以下條件設計，觀察模型的影響：

- 1. without/with load weight and with the same learning rate

=> 在同樣的lr下，有沒有已訓練好的權重，並觀察其差異性

- 2. with(without) load weight and different learning rate

=> 在上述實驗中，選擇較好的條件下，去變化lr，觀察其差異性

=> $lr = 0.1e-2, 0.1e-5$

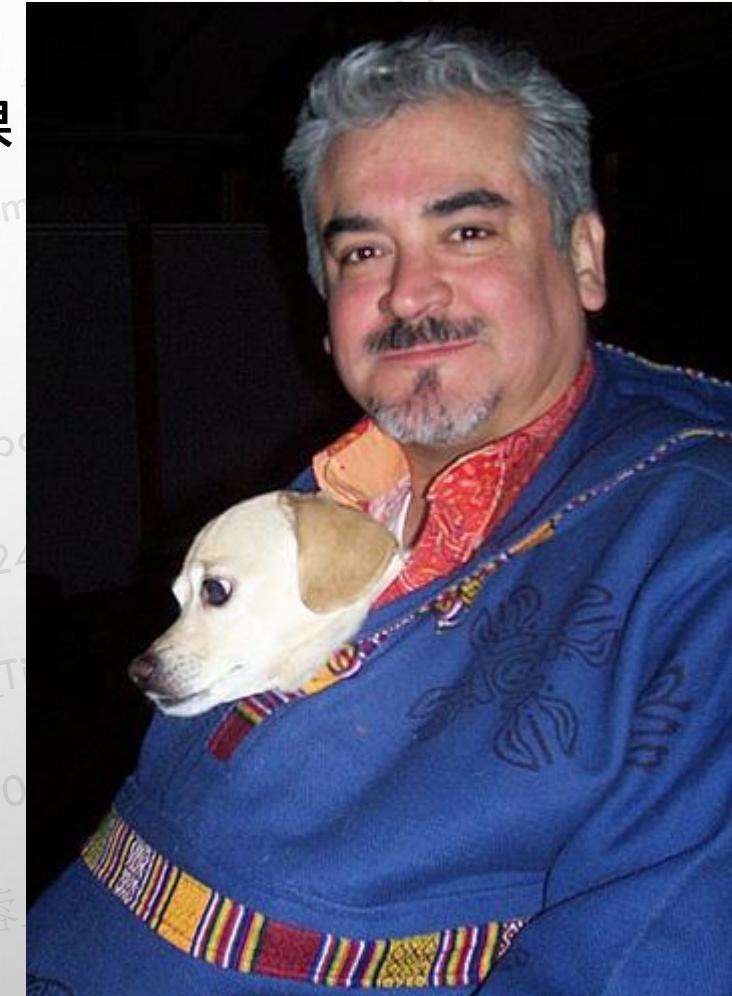
TEST 1

在不使用darknet_19的權重下，固定learning rate，觀察其學習結果

- Lr= 0.1e-3
- Epochs_start==0
- Epochs=10(google colab沒那麼多資源)
- conf_thresh=0.1,nms_thresh=0.5
- mAP=0.011

```
[9] lr = 1e-05 batch_loss = 5.----- train_batch=204
[9] lr = 1e-05 batch_loss = 4.----- train_batch=205
[9] lr = 1e-05 batch_loss = 6.----- train_batch=206
[9] lr = 1e-05 batch_loss = 6.----- train_batch=207
[9] lr = 1e-05 batch_loss = 4.----- train_batch=208
[9] lr = 1e-05 batch_loss = 4.

Evaluate ~~~~~
100% [██████████] | 4952/4952 [02:30<00:00, 32.87it/s]
Begin to perform mAP estimation
mean ap : 0.009 , best ap: 0.011
0.010722260962433904
```

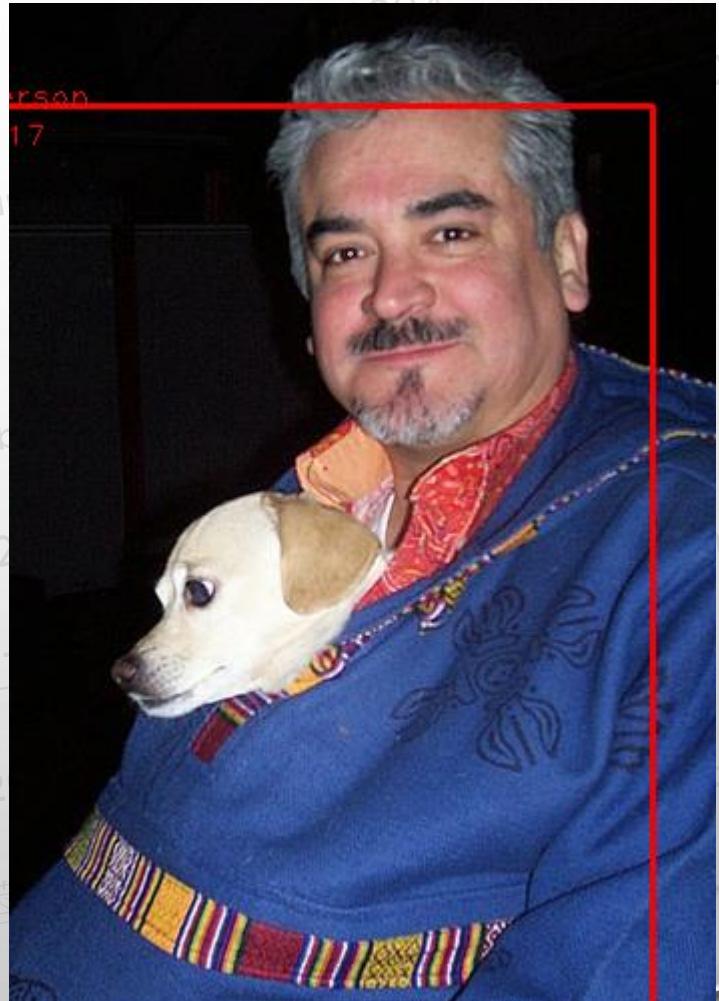


TEST 1

在使用darknet_19的權重下，固定learning rate，觀察其學習結果

- Lr= 0.1e-3
- Epochs_start==1
- Epochs=11(google colab沒那麼多資源)
- conf_thresh=0.1,nms_thresh=0.5
- mAP=0.065

```
[10] lr = 1e-05 batch_loss = 4----- train_batch=205
[10] lr = 1e-05 batch_loss = 5----- train_batch=206
[10] lr = 1e-05 batch_loss = 3----- train_batch=207
[10] lr = 1e-05 batch_loss = 1----- train_batch=208
[10] lr = 1e-05 batch_loss = 1
Evaluate ~~~~~
100%|██████████| 4952/4952 [02:43<00:00, 30.32it/s]
Begin to perform mAP estimation
mean ap : 0.036 , best ap: 0.065
0.06470927824468717
```

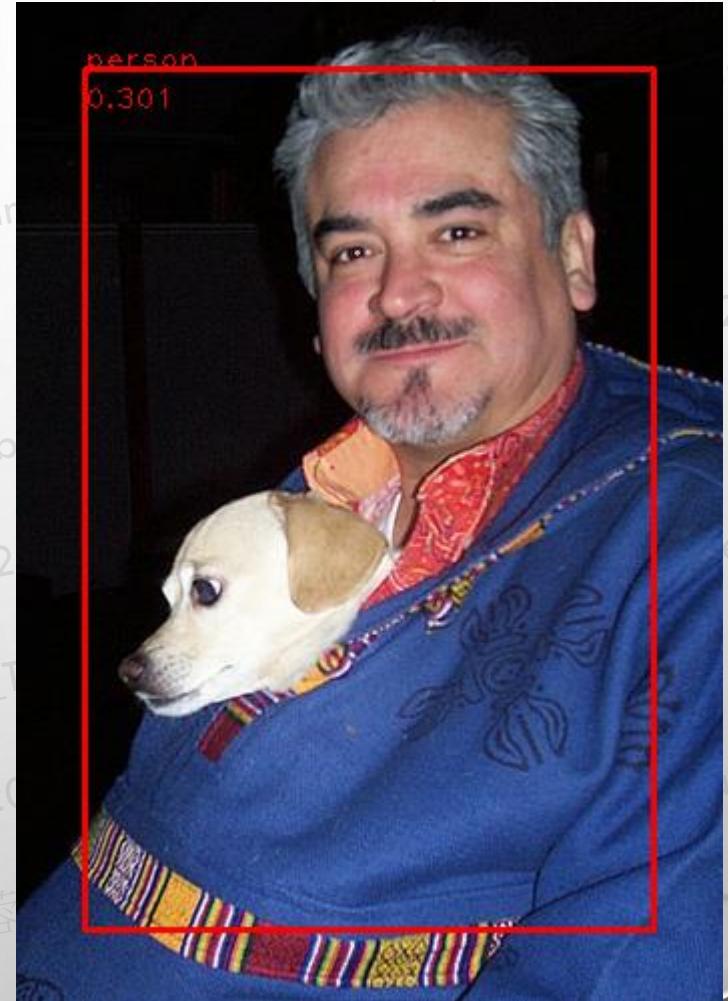


TEST 2

在使用darknet_19的權重下，固定learning rate，觀察其學習結果

- Lr= 0.1e-2
- Epochs=10(google colab沒那麼多資源)
- conf_thresh=0.1,nms_thresh=0.5
- mAP=0.141

```
[10] lr = 0.0001 batch_loss = ----- train_batch=205
----- train_batch=206
[10] lr = 0.0001 batch_loss = ----- train_batch=207
[10] lr = 0.0001 batch_loss = ----- train_batch=208
[10] lr = 0.0001 batch_loss =
Evaluate ~~~~  
100% [███████████] 4952/4952 [02:35<00:00, 31.78it/s]
Begin to perform mAP estimation
mean ap : 0.141 , best ap: 0.141
0.14127967924218646
```

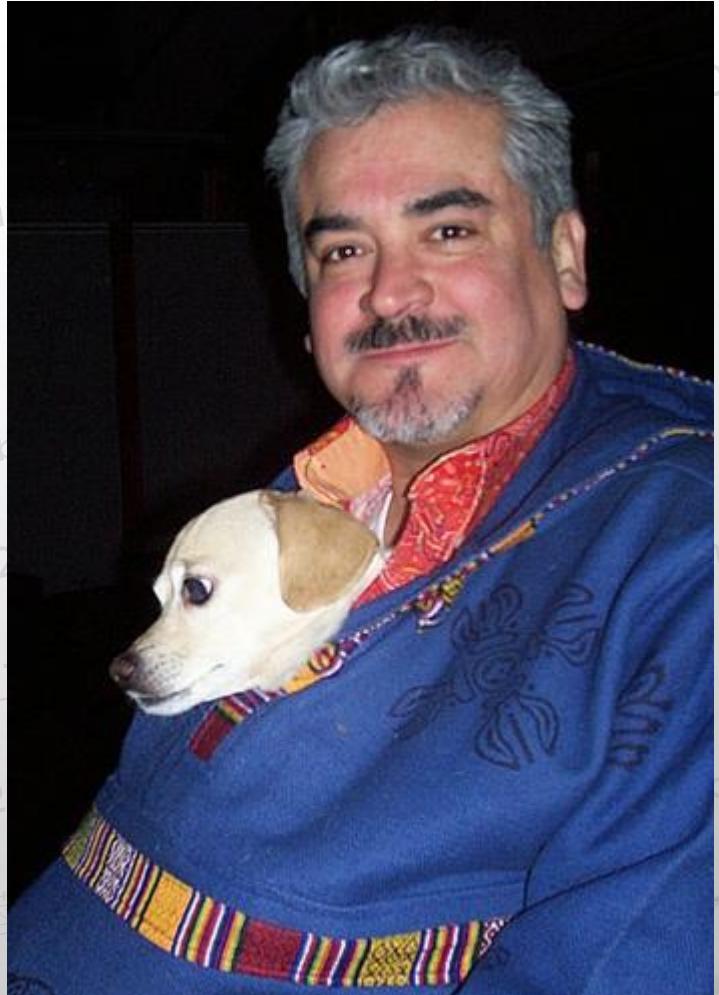


TEST 2

在使用darknet_19的權重下，固定learning rate，觀察其學習結果

- Lr= 0.1e-5
- Epochs=10(google colab沒那麼多資源)
- conf_thresh=0.1,nms_thresh=0.5
- mAP=0.006

```
[10] lr = 1e-07 batch_loss = 8----- train_batch=204
[10] lr = 1e-07 batch_loss = 8----- train_batch=205
[10] lr = 1e-07 batch_loss = 9----- train_batch=206
[10] lr = 1e-07 batch_loss = 8----- train_batch=207
[10] lr = 1e-07 batch_loss = 1----- train_batch=208
[10] lr = 1e-07 batch_loss = 8
Evaluate
100%|██████████| 4952/4952 [03:31<00:00, 23.42it/s]
Begin to perform mAP estimation
mean ap : 0.006 , best ap: 0.006
0.005724311942923993
```



CONCLUSION

- 在不使用darknet_19的權重下，mAP最好的情況只有0.011，與有darknet_19權重的條件下，mAP可以到達0.065，差異非常大。故判斷為，使用darknet_19的權重，在此狀況下能提高模型效率。
- 根據test2的測試，透過不同的learning rate，去觀察其對模型訓練的影響。結果顯示當lr過小的時候，在經過10次epochs，mAP也只有0.006，效率非常的差。而在lr較大，會呈現出其模型的極限，在短短的10次epochs，無法馬上看出效果。除外也觀察到，在 $lr > 0.01$ ，會因為梯度問題，報錯。