Turn 口罩?!

Group 3
110034560王勇盛
110034563 許宇韶
108005522 包中南
110034546 胡瑜茹

目錄

COVID-19

2020年初全球爆發COVID-19疫情，造成重大傷亡與損失，也改變了人們生活的習慣，為避免疫情的傳播，同時也保護自我健康，口罩的使用已融入我們生活之中。近期Omicron出現使指揮中心再次呼籲，民眾應落實佩戴口罩等個人防護措施

背景說明

COVID-19

問題描述 5W1H

What	你戴口罩了嗎?你保持社交距離了嗎?
When	每一秒
Who	每一個人
Where	每一個地方
Why	落實防疫政策
How	深度學習、影像處理

資料介紹、處理

COVID-19

由Kaggle公開數據集中取得Face Mask Detection的圖像資料集，資料集共分成3種類別，作為辨別該圖是否符合社交距離。

3種類別

- 帶口罩
- 不戴口罩
- 口罩佩戴不正確

由Kaggle公開數據集中取得Face Mask Detection ~12K Images Dataset的圖像資料集，分成3個資料集，作為判斷該圖是否戴口罩

3種資料集

- 訓練集資料
- 驗證集資料
- 測試集資料

資料集共分成 2 種類別

- 帶口罩
- 無戴口罩

資料處理

資料的正規化
RGB通道都是0~255的，所以我們將這個值定為0~1之間的數。

進行隨機水平翻轉
隨機的對圖片進行水平翻轉

```
train_datagen = ImageDataGenerator(rescale=1.0/255,
                                   horizontal_flip=True,
                                   zoom_range=0.2,
                                   shear_range=0.2)
train_generator = train_datagen.flow_from_directory(directory=train_dir,
                                                    target_size=(128,128),
                                                    class_mode='categorical',
                                                    batch_size=32)
```

偏移(剪切強度)
是用來進行剪切變換的程度

放大縮小
[lower,upper] = [1 - zoom_range, 1+zoom_range]

模型介紹

COVID-19

Haar Feature-based Cascade Classifier

Paul Viola 和 Michael Jones

Rapid Object Detection using a Boosted Cascade of Simple Features

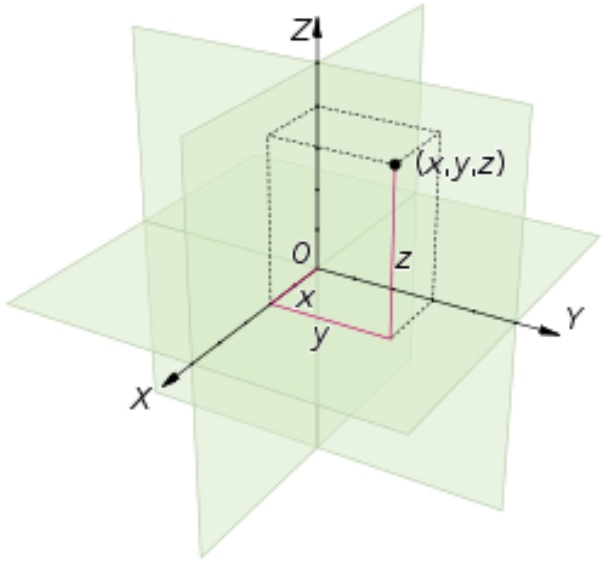
正樣本(與真實標籤一致)

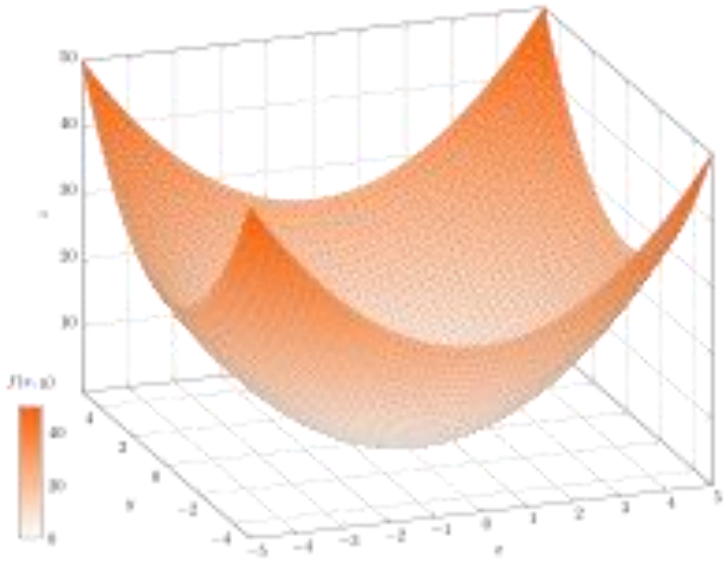
負樣本(與真實標籤不一致)

獲得圖像中人臉的邊界框坐標

Haar Cascade

模型介紹





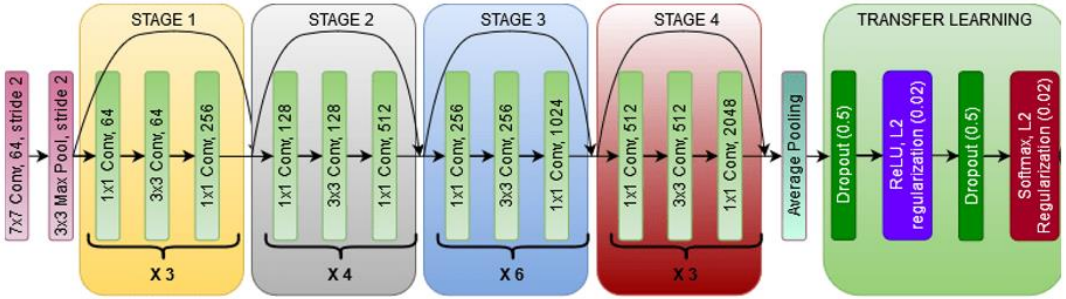
Euclidean

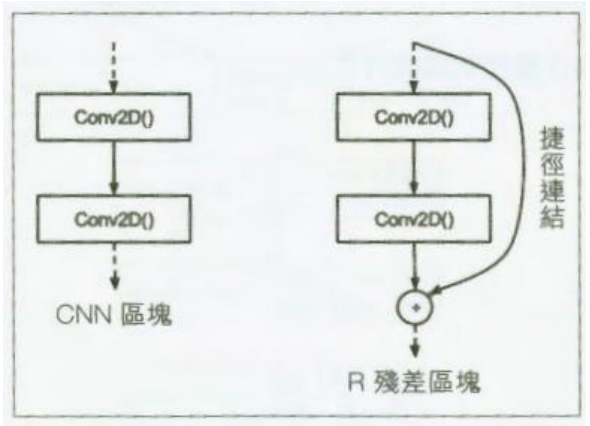
模型介紹

COVID-19

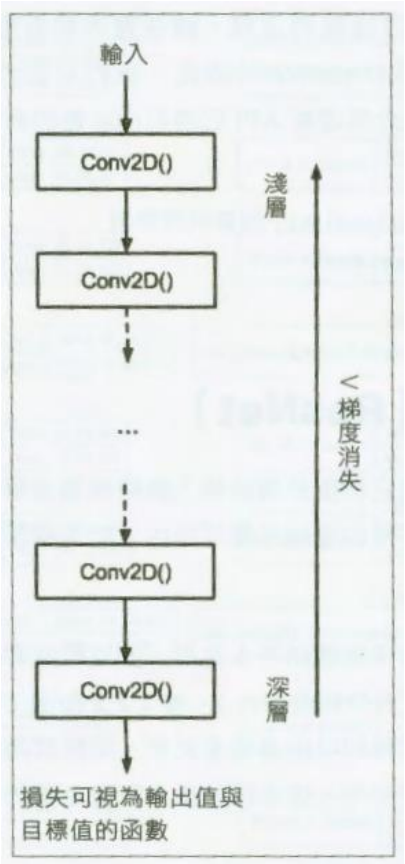
Resnet50

Resnet為深度殘差網路，在2015年的ILSVRC大賽中得分類工作的第一名。透過殘差網路解構，可以解決梯度消失的問題。





一般CNN與Resnet的區塊比較



深度網路常見問題之一

模型介紹

VGG19

在VGG中，使用了3個3x3卷積核來代替7x7卷積核，使用了2個3x3卷積核來代替5*5卷積核，這樣做的主要目的是在保證具有相同的條件下，提升了網絡的深度，一定程度上提升了神經網絡的效果。

優點：

VGGNet的結構非常簡潔，整個網絡都使用了同樣大小的卷積核尺寸（3x3）和最大池化尺寸（2x2）。

幾個小濾波器（3x3）卷積層的組合比一個大濾波器（5x5或7x7）卷積層好：驗證了通過不斷加深網絡結構可以提升性能。

缺點：

VGG耗費更多計算資源，並且使用了更多的參數，導致更多的內存佔用。

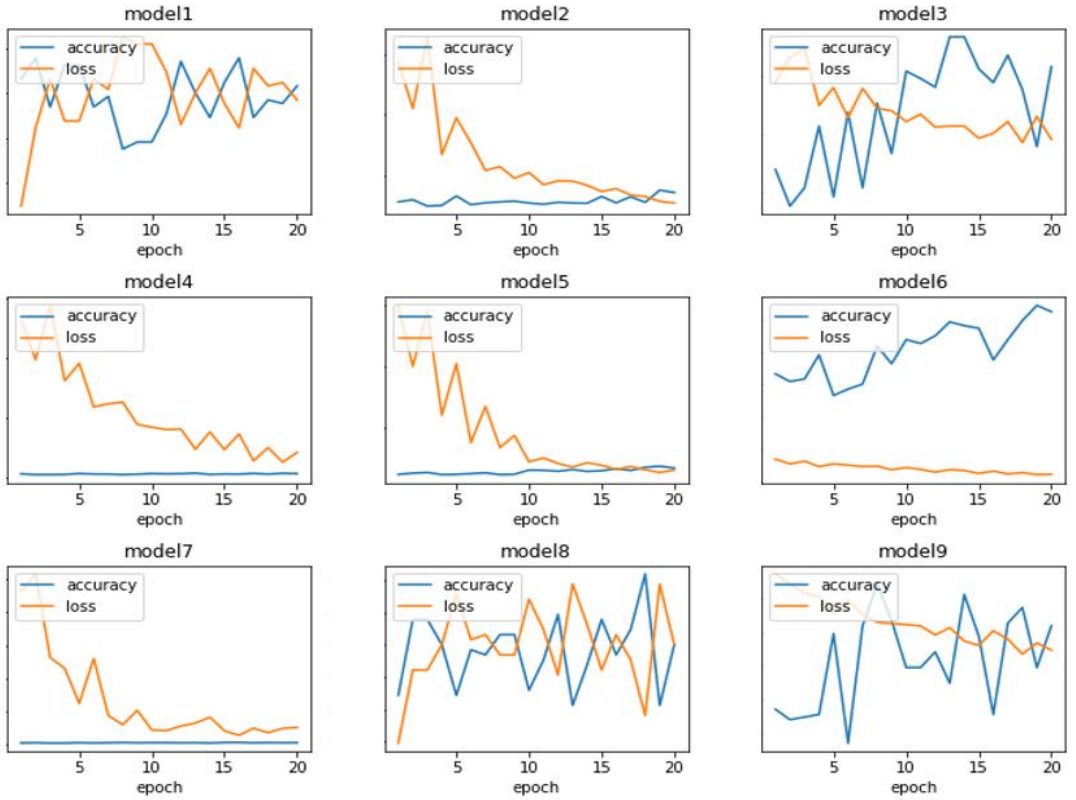
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

訓練過程

COVID-19

Model	Optimizers	Learning Rate	Loss
1 2 3	SGD	0.1 0.01 0.001	mean_squared_error binary_crossentropy categorical_crossentropy
4 5 6	Adagrad	0.1 0.01 0.001	binary_crossentropy categorical_crossentropy mean_squared_error
7 8 9	Adam	0.1 0.01 0.001	categorical_crossentropy mean_squared_error binary_crossentropy

Resnet50



訓練過程

Resnet50

Resnet50迭代20次結果

```
k = np.array(historys[6].history['accuracy'])
k
```

```
array([0.53125   , 0.5078125 , 0.515625  , 0.58984375, 0.46484375,
       0.484375  , 0.5       , 0.6171875 , 0.5625    , 0.63671875,
       0.625     , 0.6484375 , 0.69140625, 0.6796875 , 0.671875  ,
       0.57421875, 0.63671875, 0.6953125 , 0.7421875 , 0.72265625])
```

Resnet50 測試集結果

```
model.evaluate(test_generator)
```

```
7/7 [==============================] - 24s 3s/step - loss: 0.1464 - accuracy: 0.8438

[0.14643320441246033, 0.84375]
```

Resnet50迭代100次結果

```
k = np.array(historys[6].history['accuracy'])
k
```

```
array([0.4765625 , 0.51171875, 0.53515625, 0.55859375, 0.50390625,
       0.5703125 , 0.51171875, 0.72265625, 0.6171875 , 0.6875    ,
       0.6796875 , 0.6953125 , 0.62109375, 0.62890625, 0.59375   ,
       0.6484375 , 0.625     , 0.59375   , 0.73046875, 0.72265625,
       0.59765625, 0.7578125 , 0.7578125 , 0.73828125, 0.62109375,
       0.765625  , 0.765625  , 0.75      , 0.65234375, 0.73046875,
       0.609375  , 0.72265625, 0.64453125, 0.74305558, 0.73046875,
       0.77734375, 0.62890625, 0.72265625, 0.76953125, 0.81640625,
       0.8203125 , 0.83203125, 0.79296875, 0.796875  , 0.796875  ,
       0.80859375, 0.828125  , 0.8671875 , 0.81640625, 0.66796875,
       0.84765625, 0.8046875 , 0.83984375, 0.80859375, 0.78125   ,
       0.80078125, 0.78125   , 0.796875  , 0.80859375, 0.828125  ,
       0.7421875 , 0.796875  , 0.83984375, 0.7890625 , 0.78515625,
       0.828125  , 0.82421875, 0.84375   , 0.88671875, 0.8125    ,
       0.80078125, 0.796875  , 0.875     , 0.86328125, 0.78515625,
       0.890625  , 0.8671875 , 0.84375   , 0.78515625, 0.796875  ,
       0.83984375, 0.82421875, 0.84765625, 0.85546875, 0.8515625 ,
       0.78515625, 0.84375   , 0.79296875, 0.84765625, 0.81640625,
       0.859375  , 0.875     , 0.859375  , 0.80078125, 0.87109375,
       0.86328125, 0.875     , 0.83203125, 0.83203125, 0.82421875])
```

訓練過程

VGG19

VGG19 模型架構

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Functional)           (None, 4, 4, 512)         20024384
_________________________________________________________________
flatten (Flatten)            (None, 8192)              0
_________________________________________________________________
dense (Dense)                (None, 2)                 16386
=================================================================
Total params: 20,040,770
Trainable params: 16,386
Non-trainable params: 20,024,384
```

VGG19 模型訓練

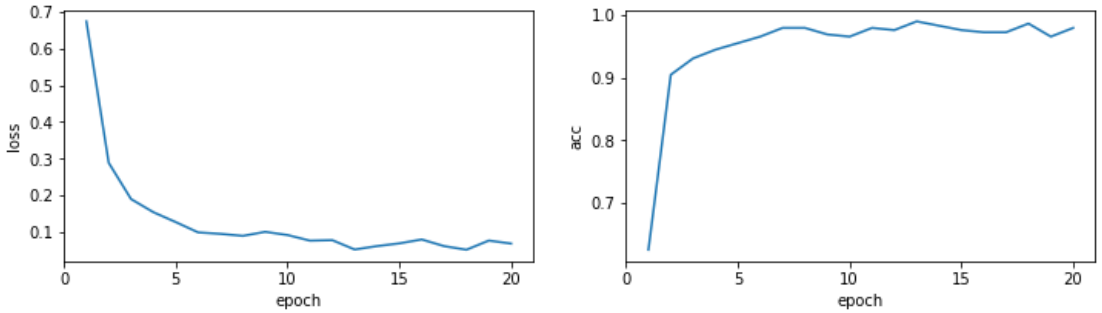
```
Epoch 1/20
9/9 [==============================] - 7s 656ms/step - loss: 0.5481 - accuracy: 0.7292
Epoch 2/20
9/9 [==============================] - 5s 543ms/step - loss: 0.2207 - accuracy: 0.9228
Epoch 3/20
9/9 [==============================] - 6s 601ms/step - loss: 0.1427 - accuracy: 0.9549
Epoch 4/20
9/9 [==============================] - 7s 735ms/step - loss: 0.1521 - accuracy: 0.9479
Epoch 5/20
9/9 [==============================] - 6s 592ms/step - loss: 0.1245 - accuracy: 0.9549
Epoch 6/20
9/9 [==============================] - 6s 608ms/step - loss: 0.0824 - accuracy: 0.9861
Epoch 7/20
9/9 [==============================] - 6s 609ms/step - loss: 0.0886 - accuracy: 0.9653
Epoch 8/20
9/9 [==============================] - 7s 726ms/step - loss: 0.0991 - accuracy: 0.9688
Epoch 9/20
9/9 [==============================] - 6s 694ms/step - loss: 0.1035 - accuracy: 0.9618
Epoch 10/20
9/9 [==============================] - 7s 763ms/step - loss: 0.0842 - accuracy: 0.9792
Epoch 11/20
9/9 [==============================] - 6s 641ms/step - loss: 0.0831 - accuracy: 0.9757
Epoch 12/20
9/9 [==============================] - 7s 622ms/step - loss: 0.0620 - accuracy: 0.9792
Epoch 13/20
9/9 [==============================] - 6s 629ms/step - loss: 0.0644 - accuracy: 0.9861
Epoch 14/20
9/9 [==============================] - 7s 717ms/step - loss: 0.0493 - accuracy: 0.9861
Epoch 15/20
9/9 [==============================] - 6s 651ms/step - loss: 0.0570 - accuracy: 0.9826
Epoch 16/20
9/9 [==============================] - 6s 662ms/step - loss: 0.0408 - accuracy: 0.9896
Epoch 17/20
9/9 [==============================] - 6s 649ms/step - loss: 0.0573 - accuracy: 0.9861
Epoch 18/20
9/9 [==============================] - 6s 672ms/step - loss: 0.0455 - accuracy: 0.9965
Epoch 19/20
9/9 [==============================] - 6s 677ms/step - loss: 0.0488 - accuracy: 0.9896
Epoch 20/20
9/9 [==============================] - 7s 709ms/step - loss: 0.0700 - accuracy: 0.9688
```

訓練過程

COVID-19

VGG19

VGG19 可視化損失函數(左)、可視化準確率(右)



VGG19 參數調整

Model	Optimizers	Learning Rate	Loss	Accuracy
1	SGD	0.1 0.01 0.001	mean_squared_error	**98.6%** 98.2% 98.1%
2	SGD	0.1 0.01 0.001	binary_crossentropy	**98.5%** 97.7% 98%
3	SGD	0.1 0.01 0.001	categorical_crossentropy	96.8% **98.3%** **98.3%**
4	Adagrad	0.1 0.01 0.001	mean_squared_error	98.8% **99.2%** 98.2%
5	Adagrad	0.1 0.01 0.001	binary_crossentropy	97.8% **98.7%** 98.3%
6	Adagrad	0.1 0.01 0.001	categorical_crossentropy	98.8% **99%** 98.7%
7	Adam	0.1 0.01 0.001	mean_squared_error	50%
8	Adam	0.1 0.01 0.001	binary_crossentropy	98.5% 98.8% **99.3%**
9	Adam	0.1 0.01 0.001	categorical_crossentropy	97.2% 98.7% **99.7%***

結論與展望

“

”

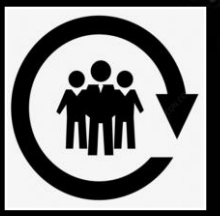
結論與展望

人臉辨識結合
門禁打卡

測量體溫

公衛安全

人流管理



疫調追蹤

THANK YOU

罩得住公司關心您的Turn口罩生活