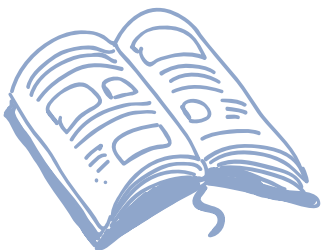


License Plate detection via ESRGAN and Faster R-CNN

Dr. Ming-Chuan Chiu

Presenter: 蔡丞洲
2020/12/24



Outline

1. 問題定義與介紹
2. 研究方法
3. 個案研究
4. 超參數調整
5. 結果分析
6. 結論

Outline

1. 問題定義與介紹
2. 研究方法
3. 個案研究
4. 超參數調整
5. 結果分析
6. 結論

背景介紹

- 背景
- 近年來，物件偵測發展快速，已經普遍出現在人類生活當中。
- 而ETC也加入了物件偵測方式來偵測車牌，並進行收取通行費，雖然分辨車牌及扣款的正確率達99.91%，但經ETC車輛眾多，目前一個月仍有4萬多筆交易處理錯誤。[1]

5W1H

WHY

- 物件偵測技術仍有改善空間。

WHAT

- 提升物件偵測準確率。

WHERE

- 高速公路的ETC裝置位置。

WHEN

- 當用戶經過ETC時。

WHO

- 用於所有經過高速公路的駕駛。

HOW

- 利用GAN生成高解析度的車牌影像，並以Faster R-CNN來訓練車牌資料庫，並使檢測準確度提高。

問題定義

- 目的：
- 近年GAN被大量的應用在AI領域當中，包含資料增強、圖片轉移、圖像合成、場景移轉等等。
- 其中GAN也可以生成出超解析度的圖像。
- 因此本研究想藉由ESRGAN生成出超解析度圖像，並透過Faster R-CNN分析，探討超解析度圖像是否提高物件偵測的準確度。

Outline

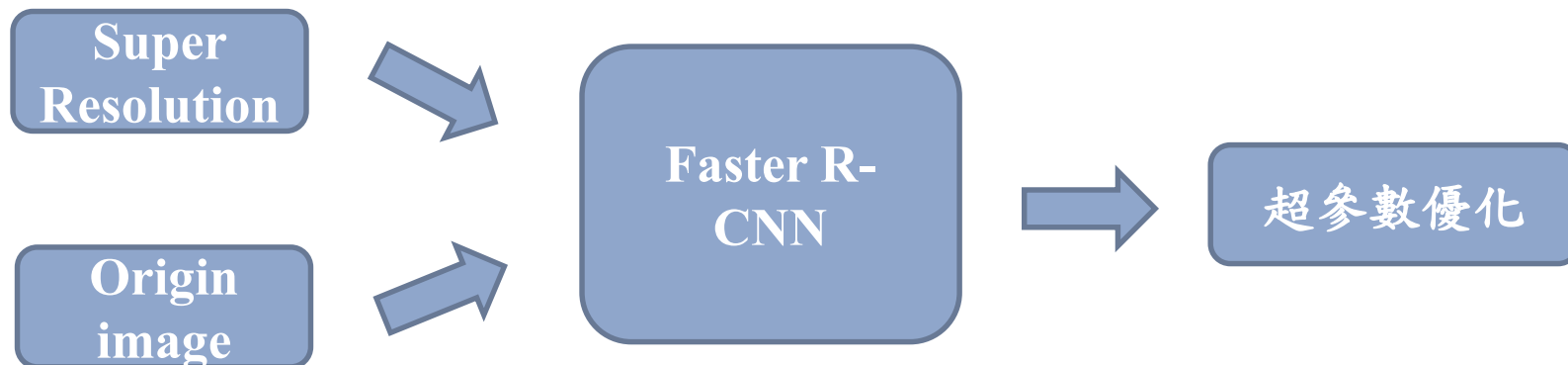
1. 問題定義與介紹
2. 研究方法
3. 個案研究
4. 超參數調整
5. 結果分析
6. 結論

研究架構

- **Stage 1**

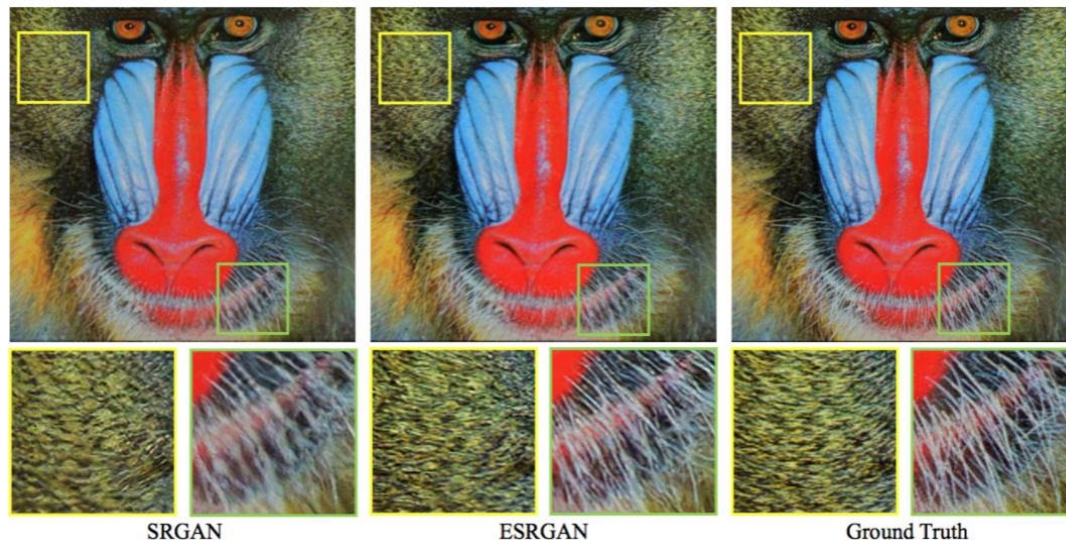


- **Stage 2**



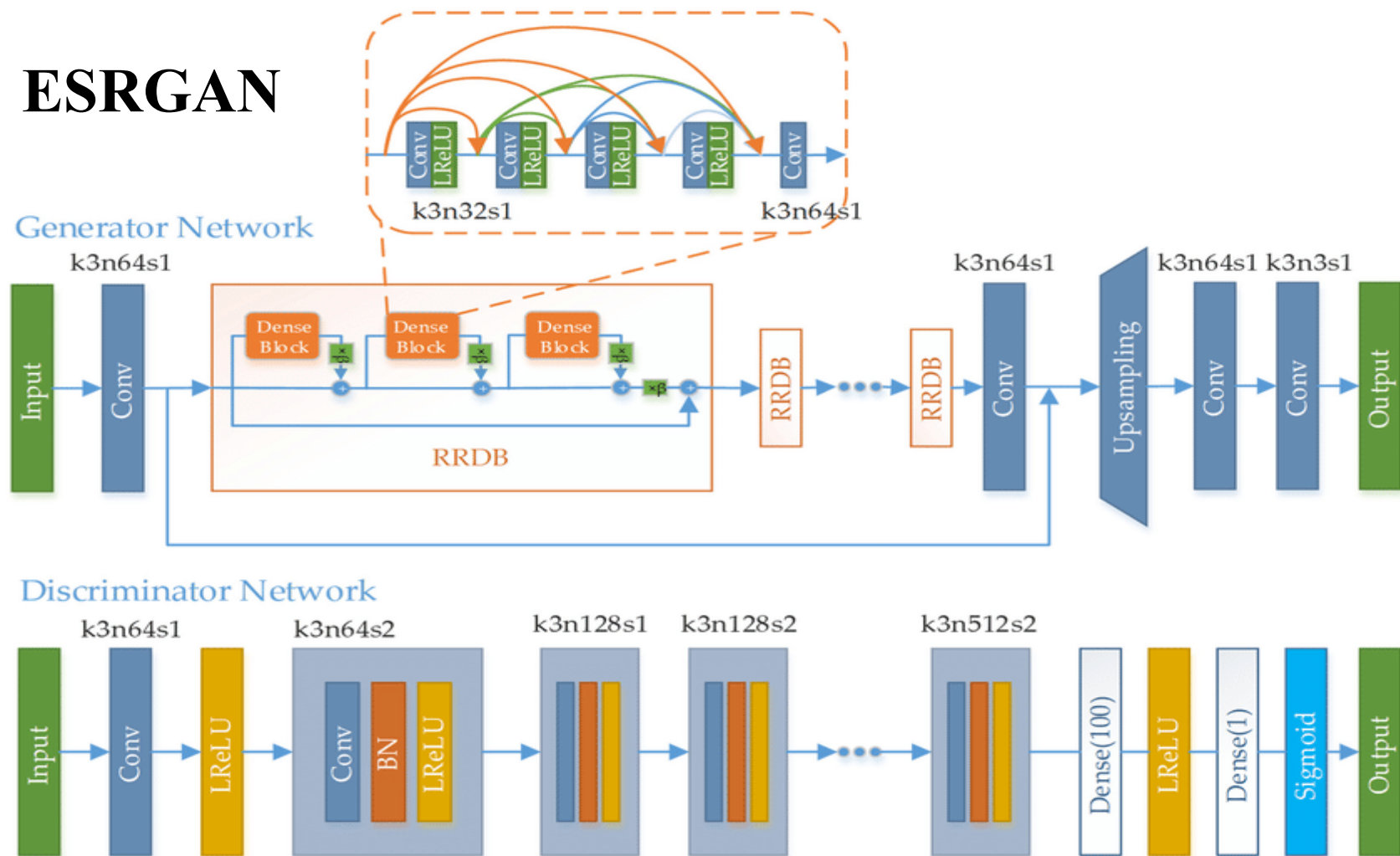
ESRGAN_(1/2)

- **ESRGAN**
 - 2018年被提出，基於修改先前的SRGAN，以提出性能更佳的模式。
 - 並且在PIRM-SR挑戰賽，贏得了第一名。



ESRGAN_(2/2)

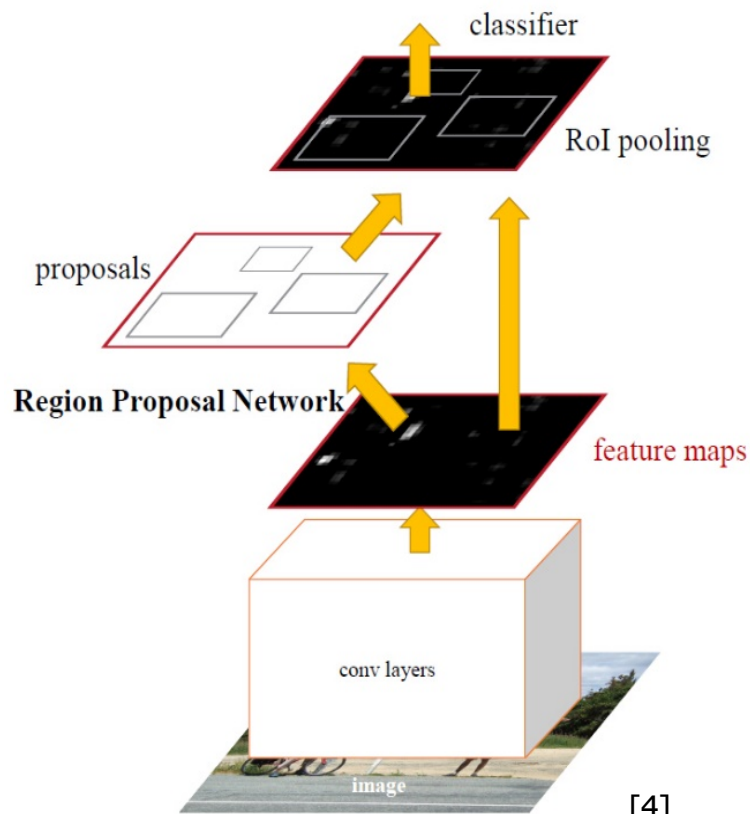
- ESRGAN



Faster R-CNN

- **Faster R-CNN**

- ✓ Convolutional layers
- ✓ Region proposal networks(RPN)
- ✓ ROI pooling
- ✓ Classification



[4]

Outline

1. 問題定義與介紹
2. 研究方法
3. 個案研究
4. 超參數調整
5. 結果分析
6. 結論

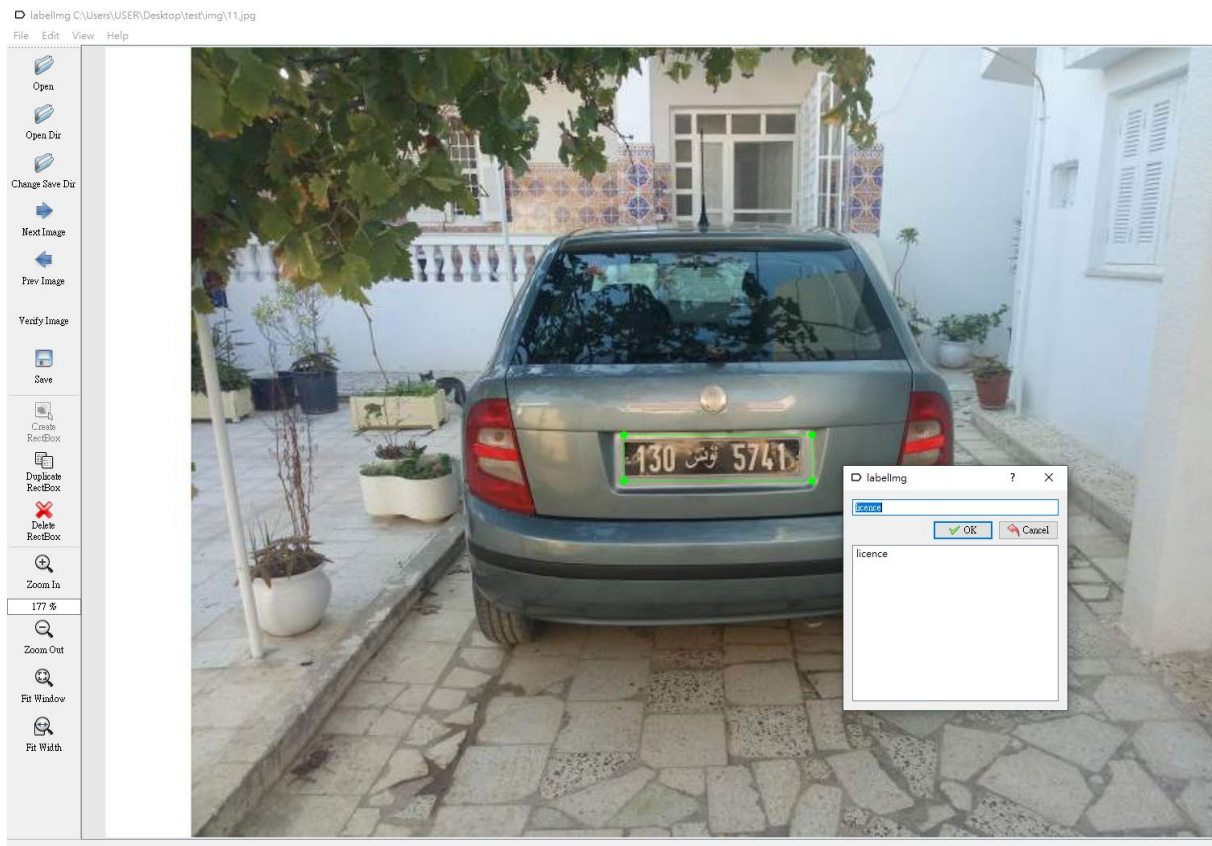
研究資料(1/2)

- 訓練集
- Dataset: Car Licence Plate Detection (kaggle)
- 853張圖像(png,xml)

- 測試集
- Dataset 1: Labeled licence plated dataset (kaggle)
- 200張圖像(jpg,xml)
- Dataset 2: Google
- 200張圖像(jpg)

研究資料(2/2)

- LabelIMG



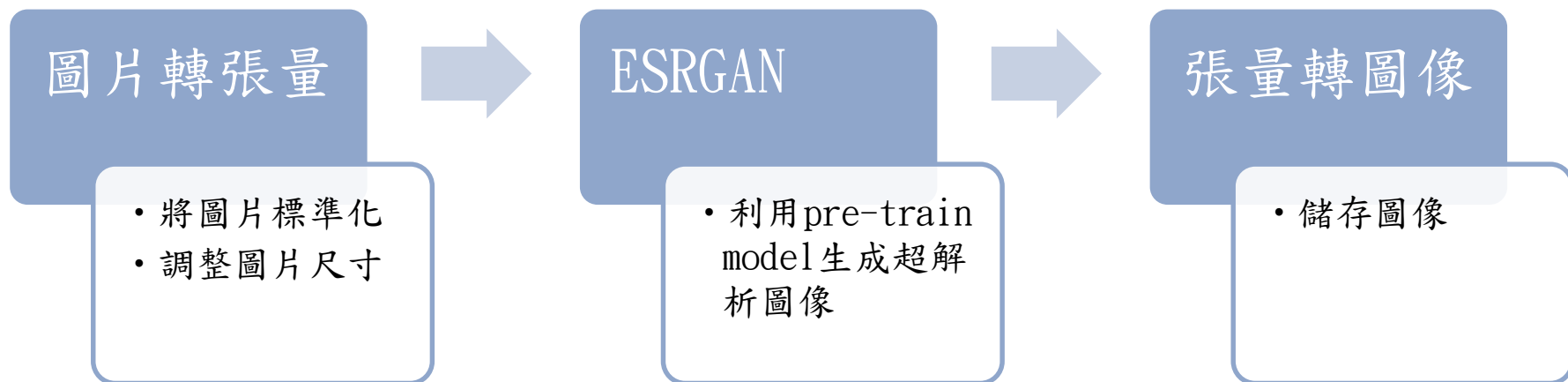
```

<annotation>
  <folder>images</folder>
  <filename>Cars4.png</filename>
  <size>
    <width>590</width>
    <height>350</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>licence</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>156</xmin>
      <ymin>82</ymin>
      <xmax>503</xmax>
      <ymax>253</ymax>
    </bndbox>
  </object>
</annotation>

```

ESRGAN (1/2)

- ESRGAN



ESRGAN (2/2)

- **ESRGAN**

```
SAVED_MODEL_PATH = "https://tfhub.dev/captain-pool/esrgan-tf2/1"  
  
model = hub.load(SAVED_MODEL_PATH)
```

Dataset Name	Link
Ground-Truth	DIV2K_train_HR
LRx4 (MATLAB bicubic)	DIV2K_train_LR_bicubic_X4

超解析圖片 (2/2)

Super Resolution

Origin



超解析圖片 (1/2)

Origin



Super Resolution

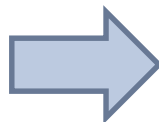


Faster R-CNN (1/3)

- Faster R-CNN(前處理)

Xml

```
<annotation>  
  <folder>images</folder>  
  <filename>Cars4.png</filename>  
  <size>  
    <width>590</width>  
    <height>350</height>  
    <depth>3</depth>  
  </size>  
  <segmented>0</segmented>  
  <object>  
    <name>licence</name>  
    <pose>Unspecified</pose>  
    <truncated>0</truncated>  
    <occluded>0</occluded>  
    <difficult>0</difficult>  
    <bndbox>  
      <xmin>156</xmin>  
      <ymin>82</ymin>  
      <xmax>503</xmax>  
      <ymax>253</ymax>  
    </bndbox>  
  </object>  
</annotation>
```



record



train.record



test.record

Faster R-CNN (2/3)

- **Faster R-CNN (faster_rcnn_inception_v2_pets.config)**
- Train path

```
train_input_reader: {  
  tf_record_input_reader {  
    input_path: "C:/tensorflow/models1/research/object_detection/train.record"  
  }  
  label_map_path: "C:/tensorflow/models1/research/object_detection/training/labelmap.pbtxt"  
}
```

- Test path

```
eval_input_reader: {  
  tf_record_input_reader {  
    input_path: "C:/tensorflow/models1/research/object_detection/test.record"  
  }  
  label_map_path: "C:/tensorflow/models1/research/object_detection/training/labelmap-test.pbtxt"  
  shuffle: false  
  num_readers: 1  
}
```

Faster R-CNN (3/3)

- Faster R-CNN (faster_rcnn_inception_v2_pets.config)**

```

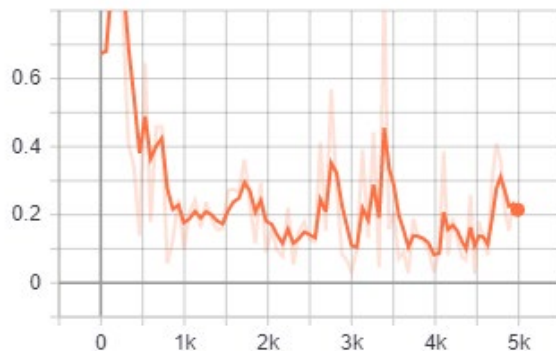
train_config: {
  batch_size: 1
  optimizer {
    momentum_optimizer: {
      learning_rate: {
        manual_step_learning_rate {
          initial_learning_rate: 0.0002
          schedule {
            step: 3000
            learning_rate: .0002
          }
          schedule {
            step: 5000
            learning_rate: .0002
          }
        }
      }
      momentum_optimizer_value: 0.9
    }
    use_moving_average: false
  }
  gradient_clipping_by_norm: 10.0
  fine_tune_checkpoint: "C:/tensorflow/models1/research/object_detection/
  faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt"
  from_detection_checkpoint: true
  load_all_detection_checkpoint_vars: true
  num_steps: 12000
  num_steps: 12000
  data_augmentation_options {
    random_horizontal_flip {

```

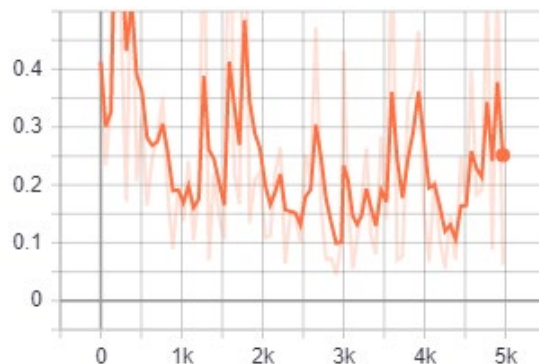
	initial
Model	Inception v2
steps	5000
Learning rate	0.0002

初步研究結果(1/2)

- **Test set 1 : Labeled licence plated dataset**
- 原圖



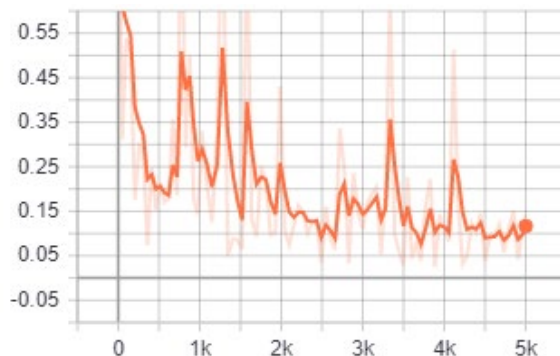
- **ESRGAN**



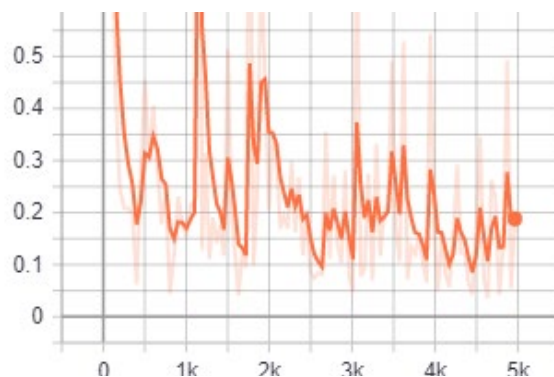
	accuracy
Origin	0.947
ESRGAN	0.961

初步研究結果(2/2)

- Test set 2 : Google
- 原圖



- ESRGAN



	accuracy
Origin	0.739
ESRGAN	0.778

Outline

1. 問題定義與介紹
2. 研究方法
3. 研究資料
4. 個案研究
5. 超參數調整
6. 結果分析
7. 結論

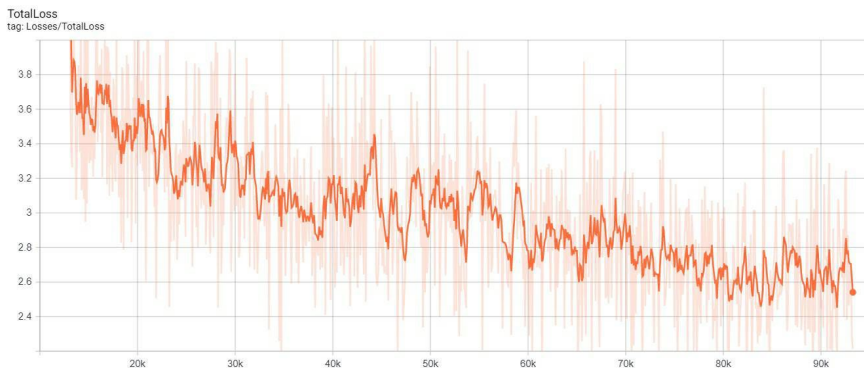
超參數調整(1/6)

- ✓ Feature extraction model
- ✓ Learning rate
- ✓ Steps
- ✓ IOU threshold

超參數調整 (2/6)

- Feature extraction model調整

models	accuracy	決策
SSD MobileNet v2	0.658	
Inception v2	0.968	✓
ResNet101 v2	0.861	



超參數調整 (3/6)

- Learning rate調整

	initial
steps	5000
Learning rate	0.0002
Iou threshold	0.6

Learning rate	accuracy	決策
0.0002	0.961	
0.0005	0.968	✓
0.0008	0.958	

超參數調整 (4/6)

- Steps調整

	initial
steps	5000
Learning rate	0.0005
Iou threshold	0.6

Steps	accuracy	決策
3000	0.951	
5000	0.961	
8000	0.968	
10000	0.972	✓
12000	0.970	

超參數調整 (5/6)

- Steps & Learning rate 調整

Steps	Learning rate	accuracy	決策
3000	0.0002	0.951	
5000	0.0002	0.961	
	0.0005	0.968	
	0.0008	0.958	
8000	0.0002	0.959	
	0.0005	0.968	
	0.0008	0.952	
10000	0.0002	0.960	
	0.0005	0.972	✓
	0.0008	0.953	

超參數調整 (6/6)

- Iou threshold調整

	initial
steps	10000
Learning rate	0.0005
Iou threshold	0.6

Iou threshold	accuracy	決策
0.7	0.971	
0.6	0.972	✓
0.5	0.957	

Outline

1. 問題定義與介紹
2. 研究方法
3. 個案研究
4. 超參數調整
5. 結果分析
6. 結論

結果分析(1/3)

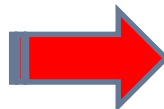
- 最終超參數組合

	超參數
Model	Inception v2
steps	10000
Learning rate	0.0005
Iou threshold	0.6

結果分析 (2/3)

- Dataset 1

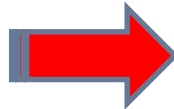
超參數調整前	accuracy
Origin	0.947
ESRGAN	0.961

1.2%

 1.1%

超參數調整後	accuracy
Origin	0.958
ESRGAN	0.972

- Dataset 2

超參數調整前	accuracy
Origin	0.739
ESRGAN	0.778

3.4%

 4.1%

超參數調整後	accuracy
Origin	0.764
ESRGAN	0.810

結果分析 (3/3)



Outline

1. 問題定義與介紹
2. 研究方法
3. 個案研究
4. 超參數調整
5. 結果分析
6. 結論

結論(1/2)

- **結論：**
- 經由ESRGAN所生成的超解析度圖像，在紋理及色彩比原始圖片加清晰。
- 使用超解析度圖片進行Faster R-CNN的檢測，能使準確度得到提升。
- 並透過超參數優化，得到準確率97.2%

結論(2/2)

- 研究限制

- 由於模型訓練時間有限：

1. 在Faster R-CNN的特徵提取模型，只使用常見的模型進行比較。
2. 沒有進行ESRGAN訓練及優化。

- 未來方向

1. 比較更多物件偵測模型來驗證超解析度圖片能提高檢測能力。
2. 利用OCR讀取車牌數字，增加研究的適用性。

參考資料

• 參考資料

- [1] <https://tisvcloud.freeway.gov.tw/>
- [2] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4681-4690).
- [3] Hicsonmez, S., Samet, N., Akbas, E., & Duygulu, P. (2020). GANILLA: Generative adversarial networks for image to illustration translation. *Image and Vision Computing*, 95, 103886.
- [4] <https://zhuanlan.zhihu.com/p/31426458>
- [5] <https://medium.com/@9821343/object-detection-api%E7%89%A9%E4%BB%B6%E8%AD%98%E5%88%A5%E5%88%86%E9%A1%9E%E5%99%A8%E5%AF%A6%E4%BD%9C-%E6%8E%A1%E7%94%A8tensorflow-cpu-%E5%9C%A8windows-10%E4%B8%8A-83d73065f27f>

Thank You for Your Listening