

# Final Report

## Image Dehazing

隊名：NTU\_b05901005\_QQ

隊員：b05901005 賴沂謙

b05901006 高珮瑋

b05901068 蕭如芸

b05901158 林緯瑋

### 1. Introduction & Motivation

天氣不佳時容易降低可見度，成為電腦視覺應用（例如object detection、classification和segmentation）的障礙。由於大多數電腦視覺模型都是在清晰的圖片上訓練，模糊的影像可能會對prediction產生不良結果。因此，Dehazing是電腦視覺應用的重要preprocessing步驟。我們實作了三篇論文的方法：Enhanced CycleGan, PFFNet和Perceptual Pyramid Deep Network。

### 2. Data Preprocessing/Feature Engineering

我們實驗做最好的結果是使用Perceptual Pyramid Deep Network的方法。這個方法中preprocess的部分，indoor和outdoor的圖片都分別做兩種處理，將兩者產生的圖片都送進model預測，再將兩者取平均作為最後的結果。Indoor和outdoor所做的處理如下：

- Indoor
  - (1) 將圖片resize成2048 x 4096的大小，再crop出5張2048 x 2048大小的圖片（由左而右，每次平移512）。
  - (2) 將圖片resize成2048 x 4096的大小，再crop出5張1024 x 4096大小的圖片（由上而下，每次平移256）。
- Outdoor
  - (1) 使用原本的圖片大小。
  - (2) 將圖片resize成3072 x 3072的大小，再crop出7張1536 x 3072大小的圖片（由上而下，每次平移256）。

### 3. Methods

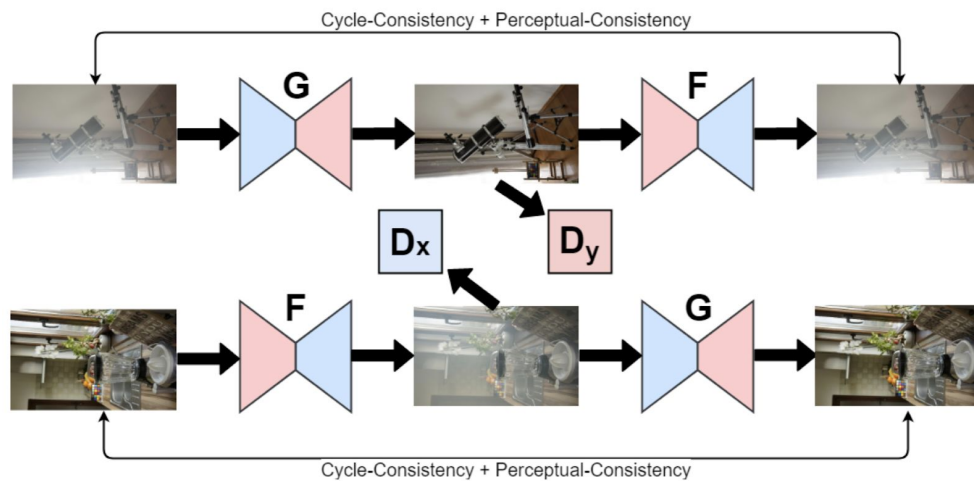
#### (1) Enhanced CycleGan

參考CycleGAN提出的方法，用於圖片到圖片轉換的cycle-consistency loss，不需要配對的數據去訓練，所以在dehazing model上，用不成對的清楚的和hazy的圖片來訓練model。

Use cycle-consistency loss and perceptual loss to compare the original image with the reconstructed cyclic-image at both feature and pixel spaces.

Use traditional Laplacian pyramid to provide better upsampling results after the main dehazing process.

模型架構：



#### (2) PFFNet

Performs feature fusion on spatial pyramid mappings between encoder and decoder, which enables maximally preserved structural details from inputs for deconvolution layers, and further makes the dehazing network more input-adaptive.

模型架構：

[Encoder]

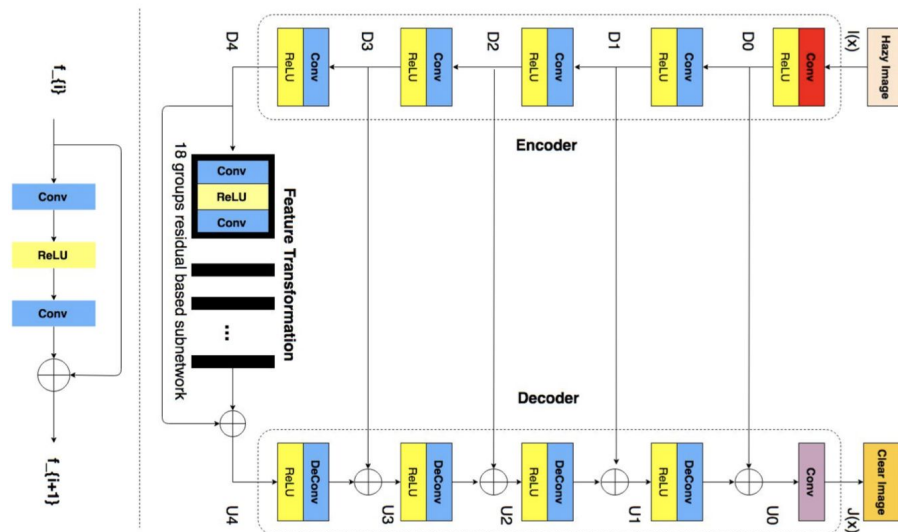
```
(sub_mean): MeanShift(3, 3, kernel_size=(1, 1), stride=(1, 1))
(add_mean): MeanShift(3, 3, kernel_size=(1, 1), stride=(1, 1))
(conv_input): ConvLayer(
  (reflection_pad): ReflectionPad2d((5, 5, 5, 5))
  (conv2d): Conv2d(3, 16, kernel_size=(11, 11), stride=(1, 1))
)
(conv2x): ConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2))
)
(conv4x): ConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2))
)
(conv8x): ConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2))
)
(conv16x): ConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2))
)
```

## [Residual Block] (共疊18個)

```
(res1): ResidualBlock(
  (conv1): ConvLayer(
    (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
    (conv2d): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
  )
  (conv2): ConvLayer(
    (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
    (conv2d): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1))
  )
  (relu): PReLU(num_parameters=1)
)
```

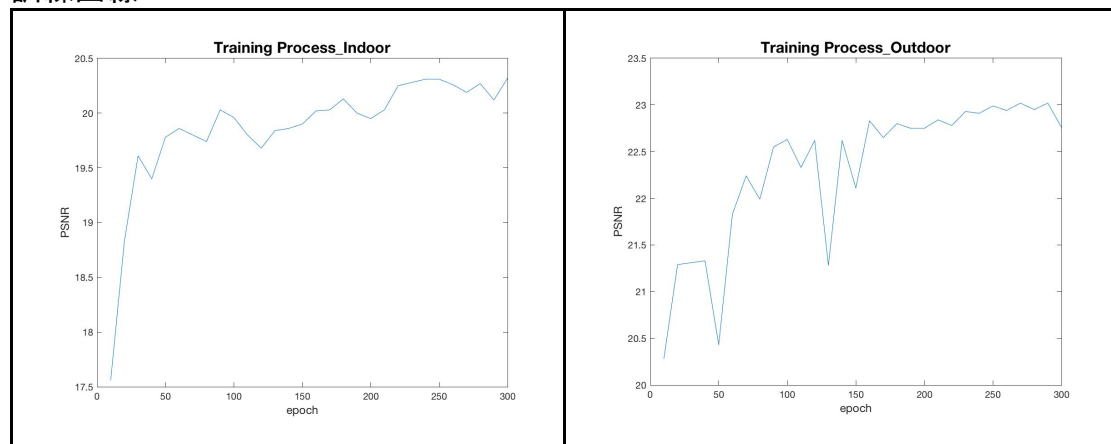
## [Decoder]

```
(convd16x): UpsampleConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): ConvTranspose2d(256, 128, kernel_size=(3, 3), stride=(2, 2))
)
(convd8x): UpsampleConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): ConvTranspose2d(128, 64, kernel_size=(3, 3), stride=(2, 2))
)
(convd4x): UpsampleConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): ConvTranspose2d(64, 32, kernel_size=(3, 3), stride=(2, 2))
)
(convd2x): UpsampleConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): ConvTranspose2d(32, 16, kernel_size=(3, 3), stride=(2, 2))
)
(conv_output): ConvLayer(
  (reflection_pad): ReflectionPad2d((1, 1, 1, 1))
  (conv2d): Conv2d(16, 3, kernel_size=(3, 3), stride=(1, 1))
)
(relu): LeakyReLU(negative_slope=0.2)
```



Loss function: Mean Square Error

訓練曲線：



### (3) Perceptual Pyramid Deep

*A multi-scale image dehazing method using Perceptual Pyramid Deep Network based on dense blocks and residual blocks.*

模型架構：

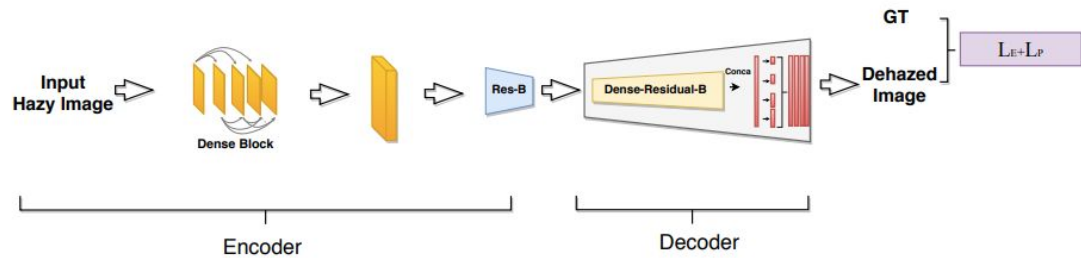
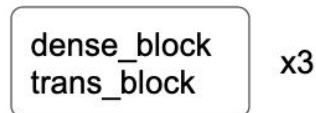


Figure 2. Overview of the proposed Multi-scale Single Image Dehazing using Perceptual Pyramid Deep Network.

Encoder:

- 參考DenseNet121:



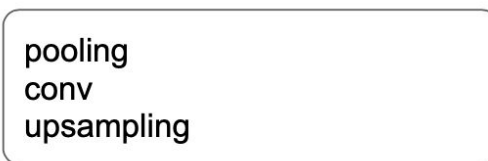
Decoder:

- 參考ResNet residual blocks的架構

dense\_residual\_block



pyramid\_pooling\_module



Loss function:

$$L = L_E + \lambda_P L_P$$

$$L_E = \frac{1}{CWH} \sum_{c=1}^C \sum_{w=1}^W \sum_{h=1}^H \|G(I(c, w, h), \Theta) - I_t(c, w, h)\|_2.$$

$$L_P = \frac{1}{C_v W_v H_v} \sum_{c=1}^{C_v} \sum_{w=1}^{W_v} \sum_{h=1}^{H_v} \|\phi_V(G(I, \Theta)) - \phi_V(I_t)\|_2$$

Multi-scale ensemble:

Multi image pyramid is used during the inference process and detection results are then combined using non-maximum suppression.

(實際實作方法如上面提到的data preprocessing)

#### 4. Experiment and Discussion : 4%

Method1: (Enhanced CycleGan)

result:

使用模型		PSNR x SSIM
Indoor	Outdoor	
pretrained model	pretrained model	<b>13.669783</b>

Method2: (PFFNet)

result:

使用模型		PSNR x SSIM
Indoor	Outdoor	
our indoor model	pretrained model	16.737262
our indoor (epoch 300)	our outdoor (epoch 290)	16.764459
pretrained	pretrained	16.802760
pretrained	our outdoor (epoch 290)	<b>16.830134</b>

Method3: (Perceptual Pyramid Deep)

result:

(1) Single Model

	使用模型		PSNR x SSIM
模型編號	Indoor	Outdoor	
1	epoch 23000	epoch 8000	19.317005

2	epoch 23000	epoch 7500	19.723085
3	epoch 23000	epoch 9000	19.785044
4	epoch 23000	epoch 9500	19.818842
5	epoch 23000	epoch 8500	<b>19.872209</b>

(2) Ensemble (模型編號如上表的Single Model)

使用模型	PSNR x SSIM
$\frac{1}{2} \times \text{模型3} + \frac{1}{2} \times \text{模型4}$	19.879351
$\frac{1}{2} \times \text{模型4} + \frac{1}{2} \times \text{模型5}$	19.908006
$\frac{1}{2} \times \text{模型5} + \frac{1}{4} \times \text{模型3} + \frac{1}{4} \times \text{模型4}$	19.911548
$\frac{1}{3} \times \text{模型5} + \frac{1}{3} \times \text{模型3} + \frac{1}{3} \times \text{模型4}$	<b>19.913006</b>




三種方法結果比較：（分別使用上述該方法中分數最高的模型）

Method	Best Score
Enhanced CycleGan	13.669783
PFFNet	16.830134
Perceptual Pyramid Deep	19.913006




Discussion:

(1) 比較三種方法中最好的結果

Enhanced CycleGan的結果，雖然有成功消除一點霧霾，但是整體的解析度卻變很差，極可能是分數最低的原因；PFFNet的結果，雖然解析度有好一點，但是不管indoor或outdoor都會出現奇怪的色塊，呈現很不自然的現象；最好的結果是用Perceptual Pyramid Deep的做法，可看出雖然indoor因為最後會將模型產生的好幾張圖片拼起來，所以會有一些長方形的拼貼痕跡，但是整體而言照片很清晰，而且也沒有像PFFNet有不自然色塊的現象，所以最後我們選擇使用Perceptual Pyramid Deep的方法繼續實作。

	Indoor : Test/4.jpg
Enhanced CycleGan	
PFFNet	
Perceptual Pyramid Deep	



	Outdoor : Test/10.jpg
Enhanced CycleGan	
PFFNet	
Perceptual Pyramid Deep	

(2) 比較第三種方法不同模型的結果

a. Indoor epoch 23000 vs 26000

epoch 23000的圖片較26000的圖片明亮一些，但其實肉眼蠻難看出兩者的差別，不過PSNR x SSIM的分數用23000的結果卻高了很多

Indoor	Test/5.jpg
--------	------------




23000	
26000	

b. Outdoor epoch 8000 vs 8500 vs 9500

取出PSNR x SSIM分數最差的epoch 8000，與分數最好的8500和第二好的9500的圖片做比較。可看出分數最差的8000照片，整體的亮度比另外兩個都來得暗許多，其中8500的照片很明顯整體亮度比其他epoch產生的照片都來得清晰與明亮，所以分數較高的結果比較不意外。

較奇怪的是epoch 8000的分數卻比epoch 7500的分數還要來得差，但在epoch 8500的時候又突然變得很高，我們推測的可能是模型快收斂時會一直波動，所以剛好在8000時存到較差的模型。

Outdoor	Test/10.jpg
8000	
8500	

9500	
------	--

## 5. Conclusion

比較三種方法的結果，發現Perceptual Pyramid Deep Network所得到的結果PSNRx SSIM分數最高，其中又以ensemble三種訓練不同epoch的模型所得到的結果最好。

## 6. Reference

- (1) Cycle-Dehaze: Enhanced CycleGAN for Single Image Dehazing

Paper: <https://arxiv.org/pdf/1805.05308.pdf>

Github: <https://github.com/engindeniz/Cycle-Dehaze>

- (2) Progressive Feature Fusion Network for Realistic Image Dehazing (PFFNet)

Paper: <https://arxiv.org/pdf/1810.02283.pdf>

Github: <https://github.com/MKFMiku/PFFNet>

- (3) Multi-scale Single Image Dehazing using Perceptual Pyramid Deep Network

Paper:

[http://openaccess.thecvf.com/content\\_cvpr\\_2018\\_workshops/papers/w13/Zhang\\_Multi-Scale\\_Single\\_Image\\_CVPR\\_2018\\_paper.pdf](http://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w13/Zhang_Multi-Scale_Single_Image_CVPR_2018_paper.pdf)

Github: <https://github.com/hezhangsprinter/NTIRE-2018-Dehazing-Challenge>