Neural Style Transfer

2022. 08. 05. AI 4.0 Studio

Presentated By

Mentor | 김현주, 양희수
Mentee | 김표재, 이호준, 정영찬

CONTENTS

Introduction

프로젝트 소개

Introduction

◀ Style Image

Content Image ▶

스타일 트랜스퍼 (Style Transfer)

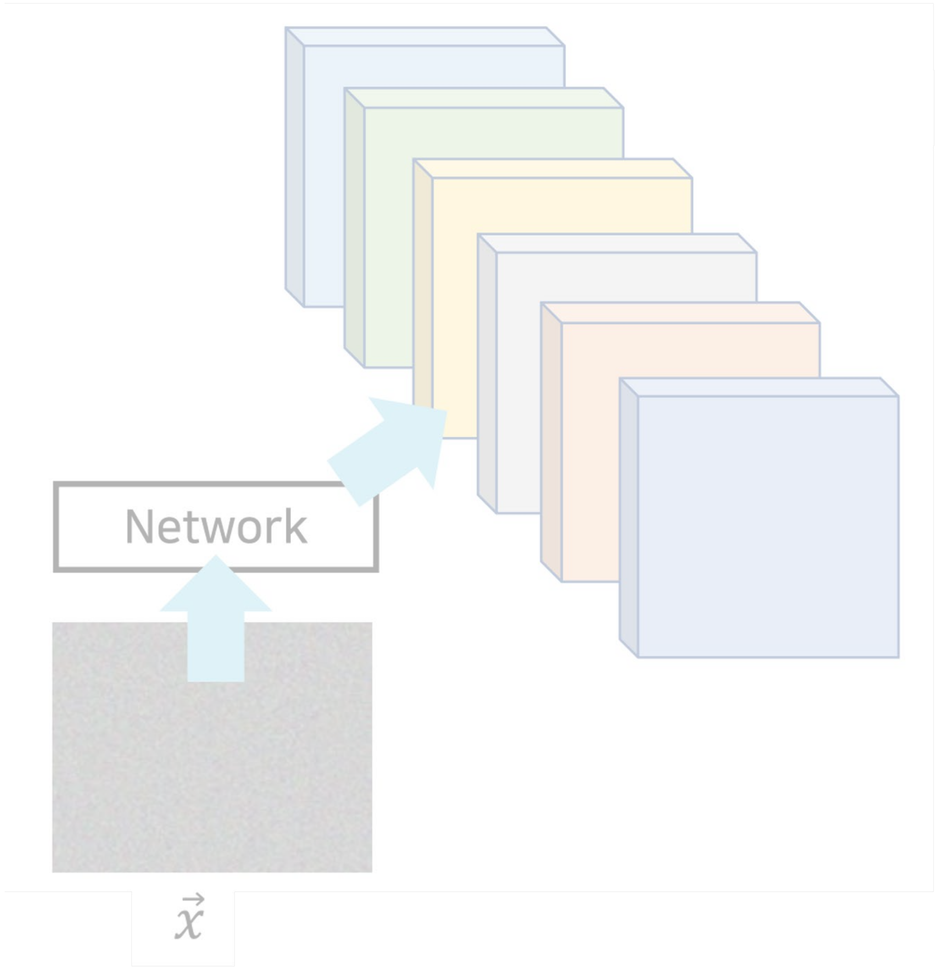
스타일 이미지(Style)가 주어졌을 때, 해당 이미지를 그대로 필터처럼 사용하는 것이 아니라, 특징(feature)을 추출하여 콘텐츠 이미지(content)에 전이시키는 기법

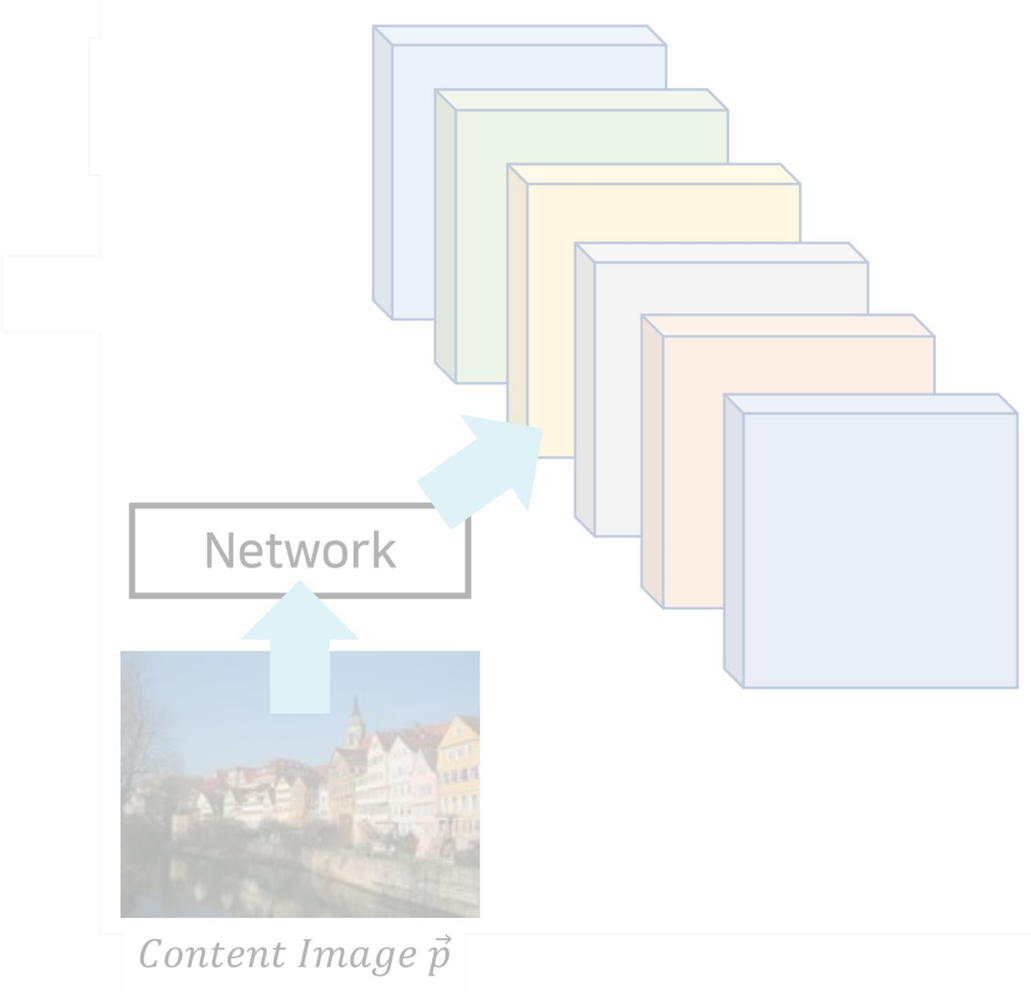
◀ Style Image

Content Image ▶

Basic Principles

기본적인 원리 소개





Basic Principles

Content image $\vec{p}$

Style Image $\vec{a}$

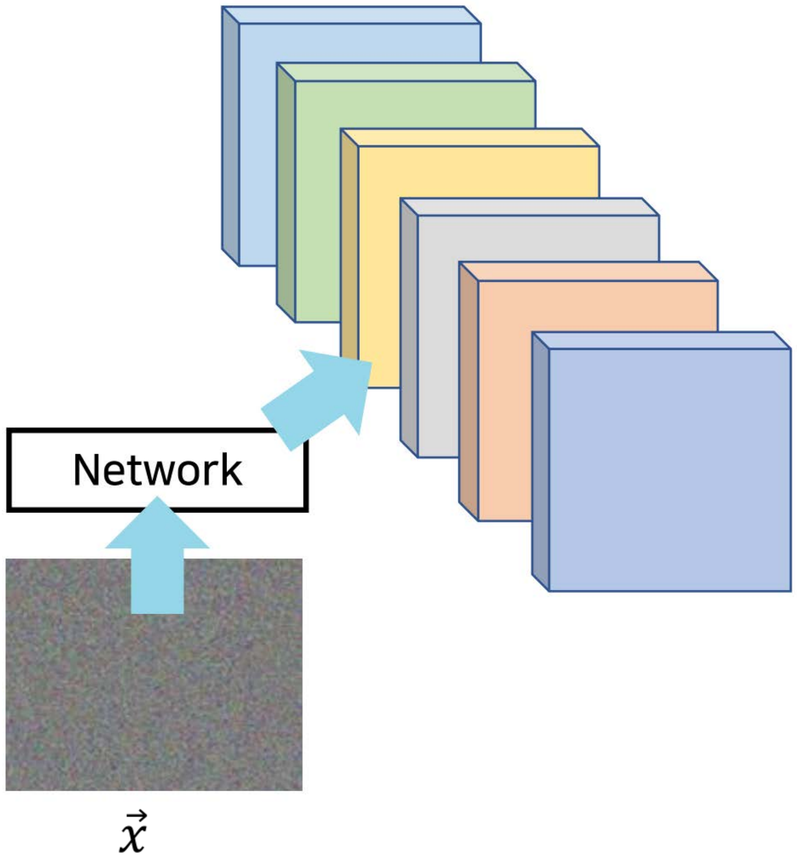
$L_{content}$

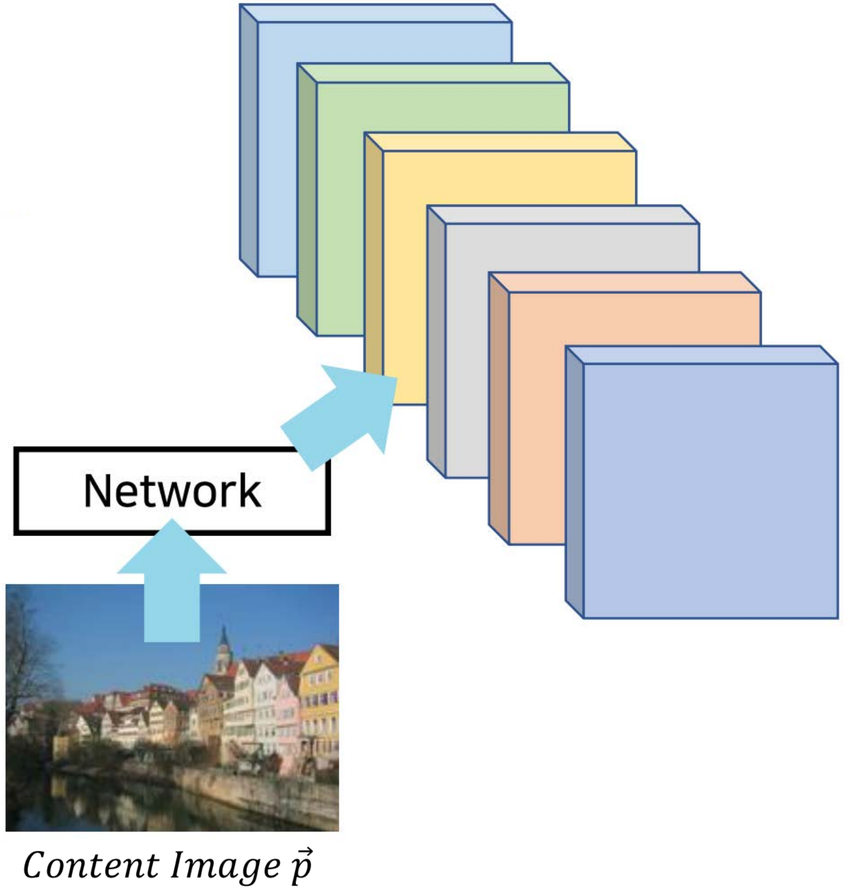
L_{style}

$\vec{x}$

$Result\ \vec{x}$

콘텐츠 손실(content loss)는 두 이미지 특징(feature)의 활성값이 동일하도록 만듭니다

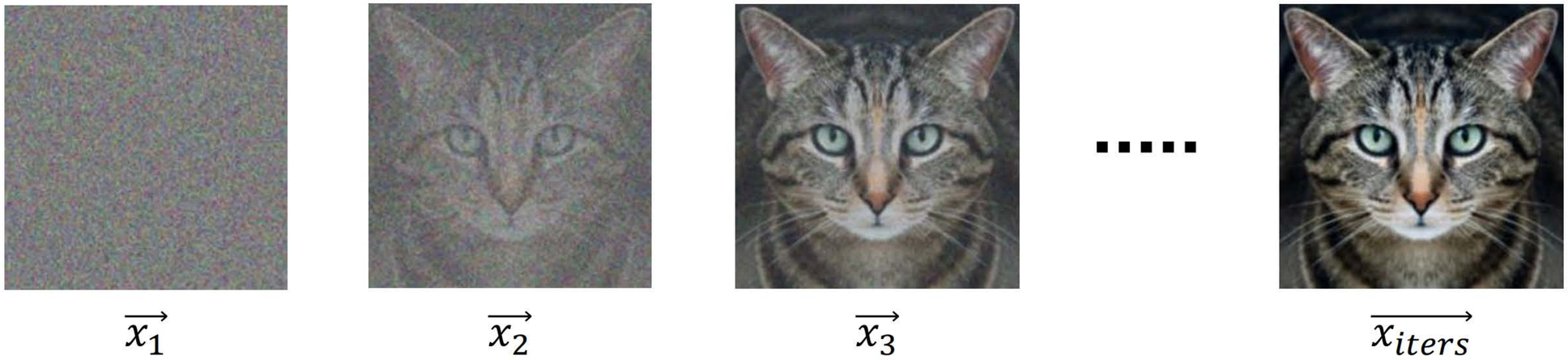




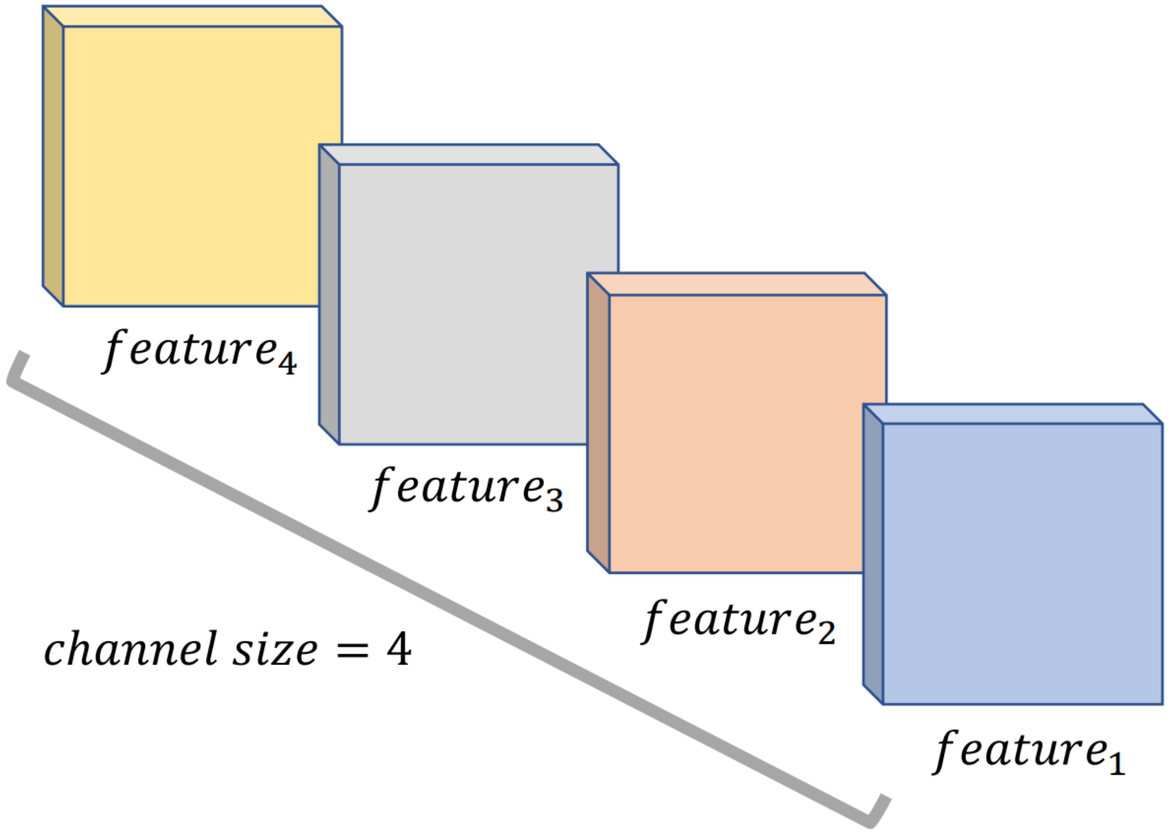
네트워크의 가중치는 고정한 뒤에 이미지를 변경시키는 방법을 사용합니다.
- 이미지를 학습(업데이트)한다고 이해할 수 있습니다.

타겟 이미지

[이미지 변수 x 값이 변환되는 과정]



Basic Principles



스타일(Style)은
서로 다른 특징(feature)간의
상관관계(correlation)을 의미합니다.

[Style Reconstruction(스타일 재구성)]

- Gram Matrix는 채널의 크기만큼 커지게 됩니다.

(a) conv1_1

(b) conv1_1, conv2_1

(c) conv1_1, conv2_1, conv3_1

(d) conv1_1, conv2_1, conv3_1, conv4_1,

(e) conv1_1, conv2_1, conv3_1, conv4_1, conv5_1

[Content Reconstruction(내용 재구성)]

- 깊어질수록 구체적인 픽셀 정보는 소실됩니다.

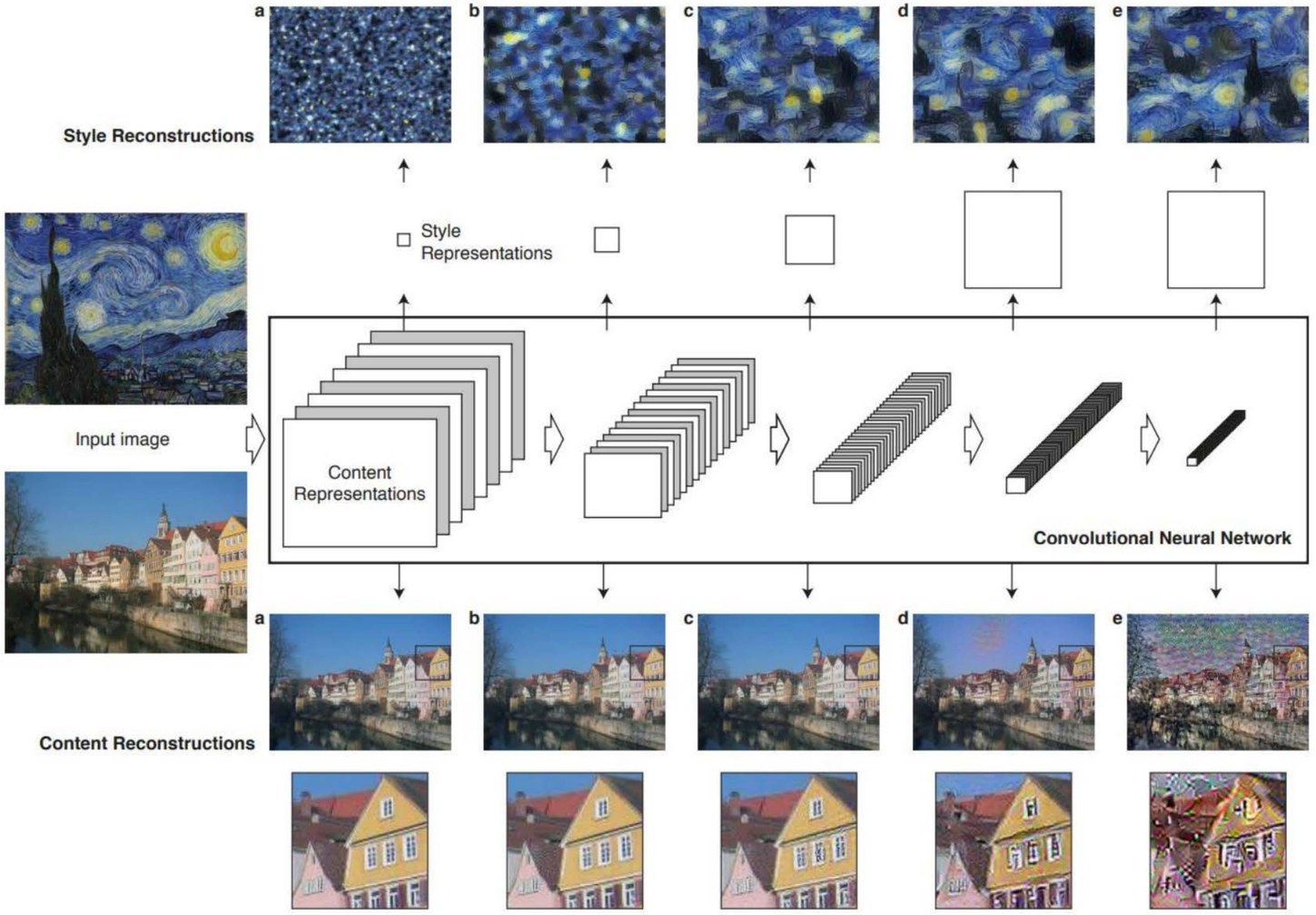
(a) conv1_2

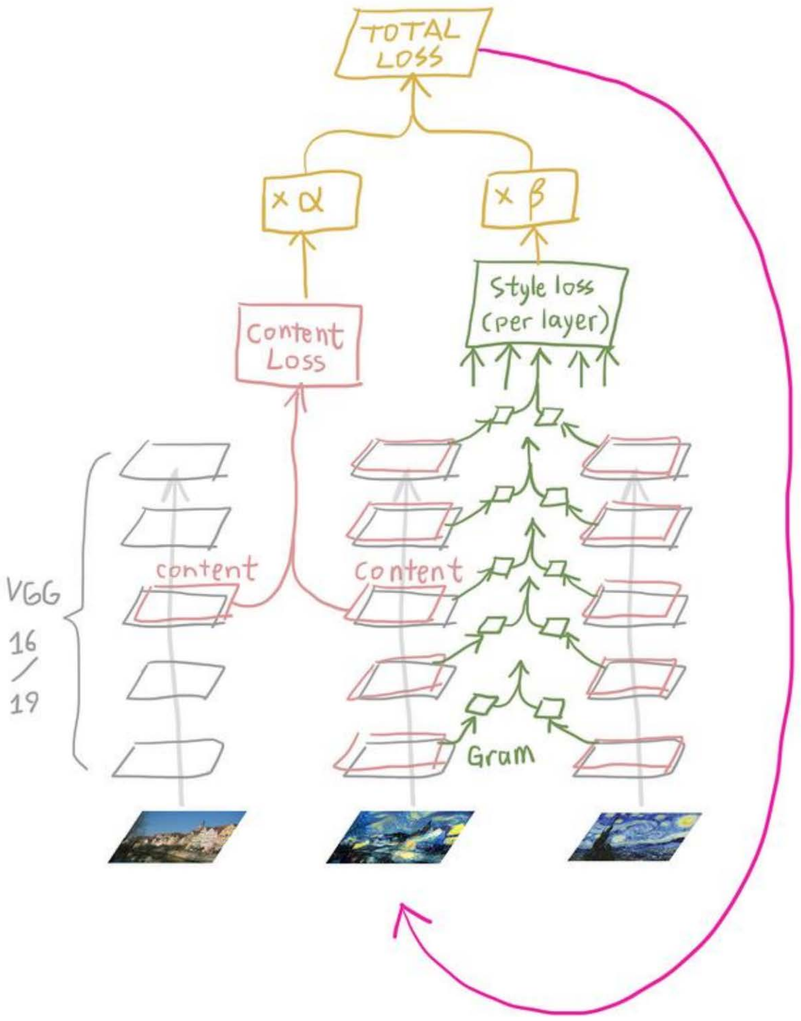
(b) conv2_2

(c) conv3_2

(d) conv4_2

(e) conv5_2





Style Transfer 알고리즘의 과정

컨텐츠 업데이트 / 스타일 업데이트

```
prediction_probabilities = vgg(x)
prediction_probabilities.shape
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels.h5
574717952/574710816 [==============================] - 3s 0us/step
574726144/574710816 [==============================] - 3s 0us/step
TensorShape([1, 1000])
```

```
[ ] predicted_top_5 = tf.keras.applications.vgg19.decode_predictions(prediction_probabilities.numpy())[0]
    [(class_name, prob) for (number, class_name, prob) in predicted_top_5]
```

```
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json
40960/35363 [==================================] - 0s 0us/step
49152/35363 [==========================================] - 0s 0us/step
[('envelope', 0.61191404),
 ('web_site', 0.20023946),
 ('binder', 0.018892692),
 ('carton', 0.014203147),
 ('switch', 0.011903387)]
```

Codes & Layers

사용한 코드와 레이어 소개

```
[ ] vgg = tf.keras.applications.VGG19(include_top=False, weights='imagenet')

    print()
    for layer in vgg.layers:
      print(layer.name)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg19/vgg19_weights_tf_dim_ordering_tf_kernels_notop.h5
80142336/80134624 [==============================] - 1s 0us/step
80150528/80134624 [==============================] - 1s 0us/step

input_2
```

초기 모델 레이어

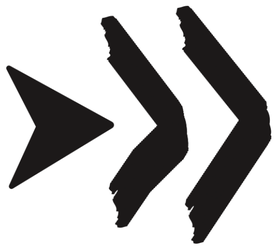
이미지 처리를 위해, 분류 레이어를 제외한 전체 레이어들을 조절하며 최적의 결과값을 찾는 과정을 거침

```
content_layers = ['block5_conv2']

style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']

num_content_layers = len(content_layers)
num_style_layers = len(style_layers)
```

초기 모델 레이어

이미지 처리를 위해, 분류 레이어를 제외한 전체 레이어들을 조절하며 최적의 결과값을 찾는 과정을 거침

```
content_layers = ['block5_conv2']

style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']

num_content_layers = len(content_layers)
num_style_layers = len(style_layers)
```

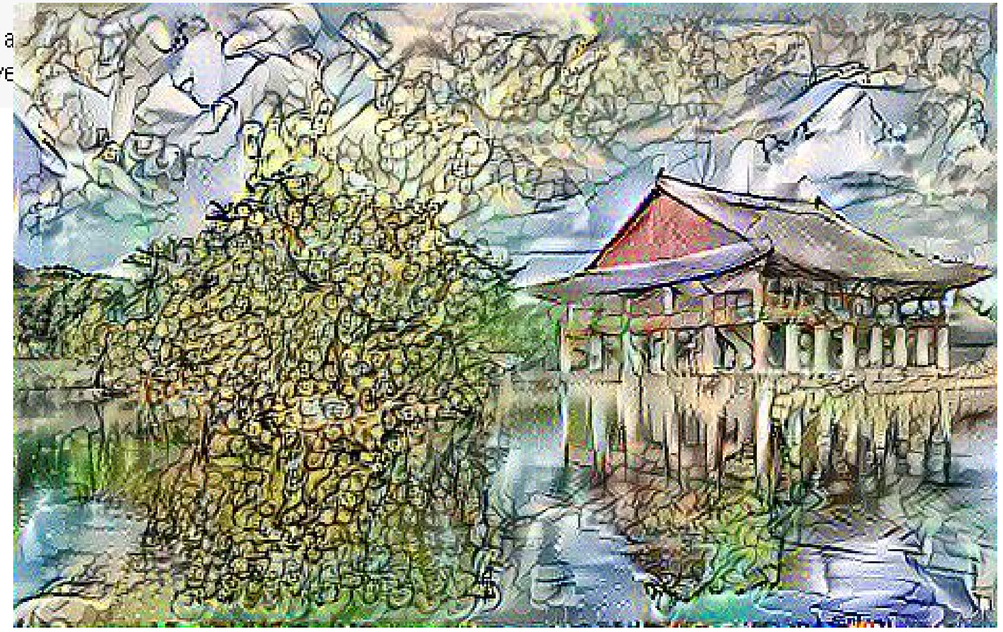


```
block1_conv1
block1_conv2
block1_pool
block2_conv1
block2_conv2
block2_pool
block3_conv1
block3_conv2
block3_conv3
block3_conv4
block3_pool
block4_conv1
block4_conv2
block4_conv3
block4_conv4
block4_pool
block5_conv1
block5_conv2
block5_conv3
block5_conv4
block5_pool
```

Codes & Layers

[초기 선택 레이어]

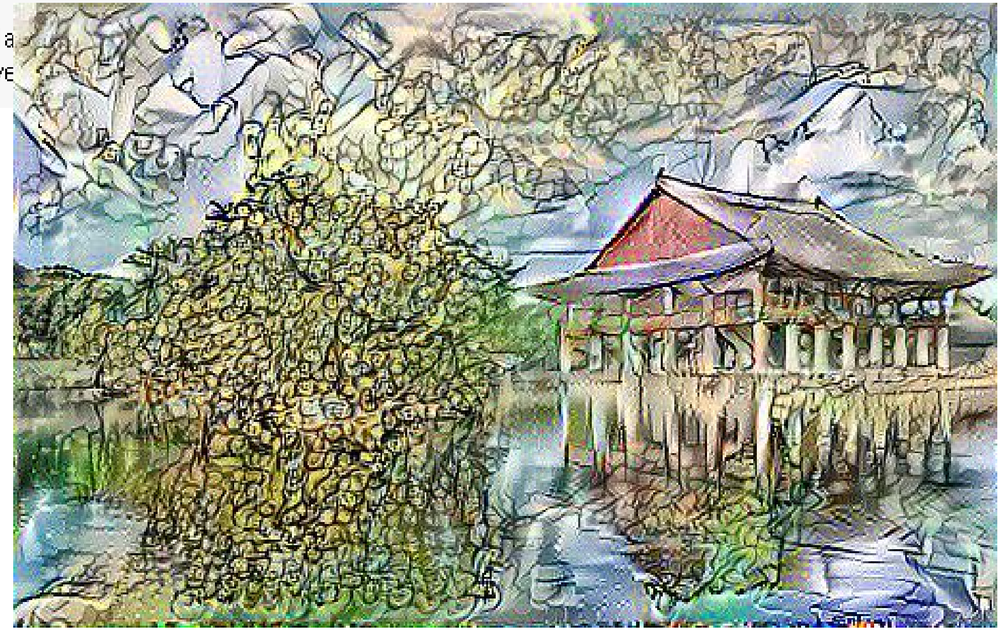
```
content_layers = ['block5_conv2']

style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']

num_content_la
num_style_laye
```



Codes & Layers

[초기 선택 레이어]

```
content_layers = ['block5_conv2']

style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']

num_content_la
num_style_laye
```

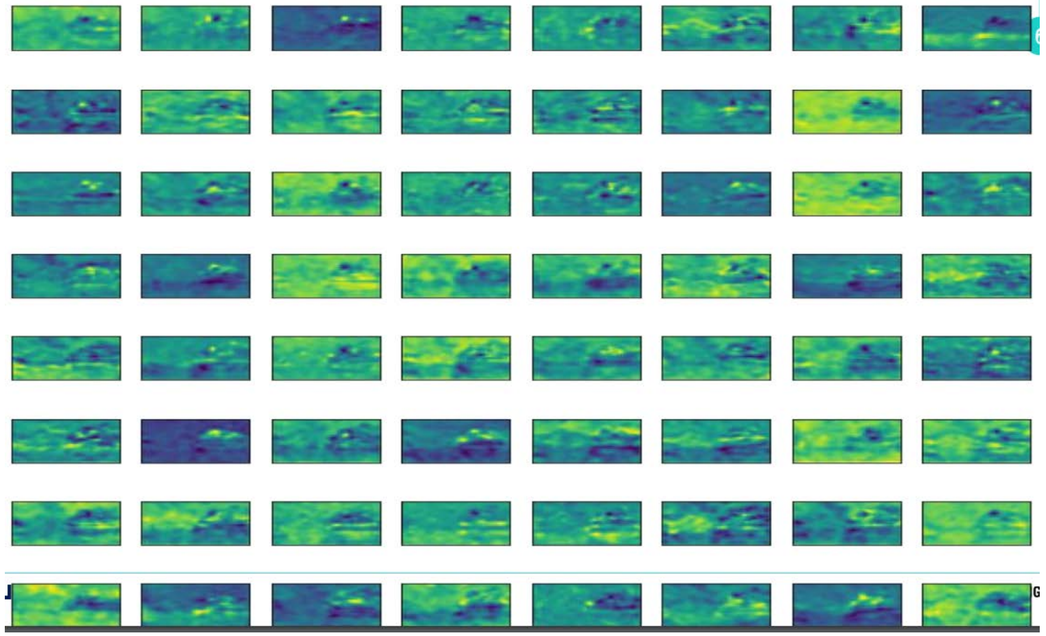


[더 얕은 콘텐츠 레이어]

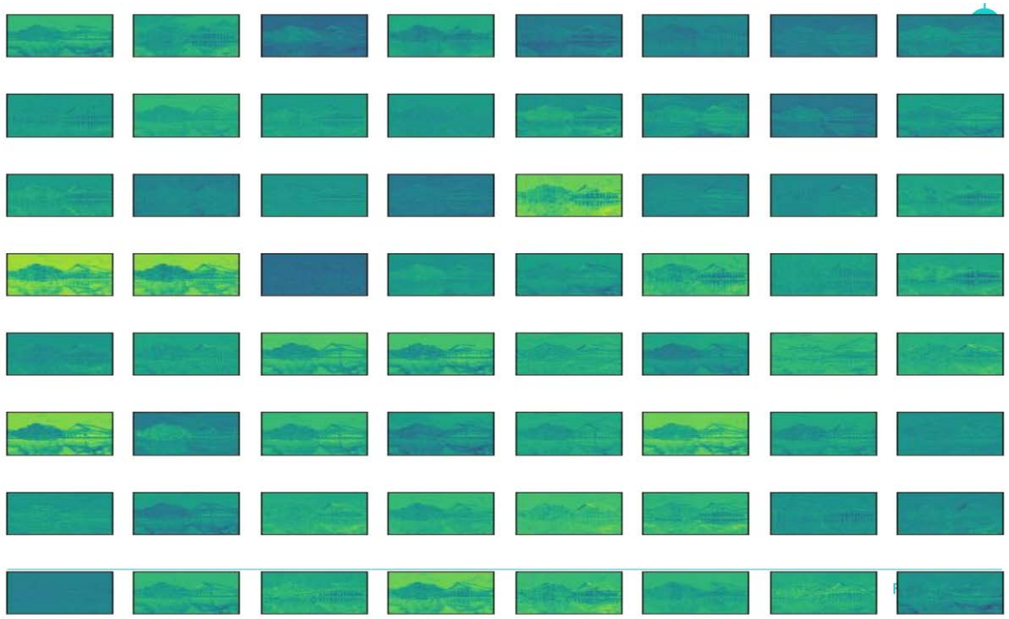
```
content_layers = ['block2_conv1']

style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']

num_content_lay
num_style_layer
```

Codes & Layers

[초기 선택 레이어]

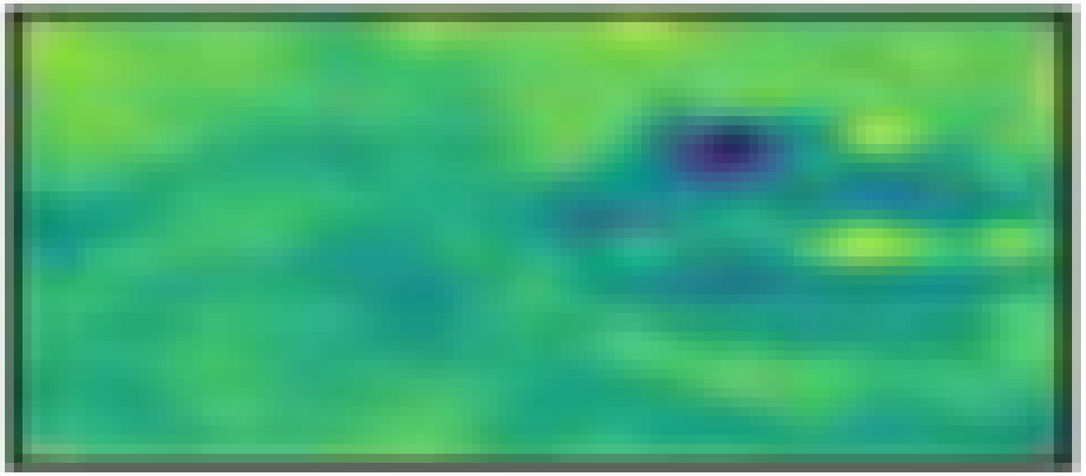


[더 얕은 콘텐츠 레이어]

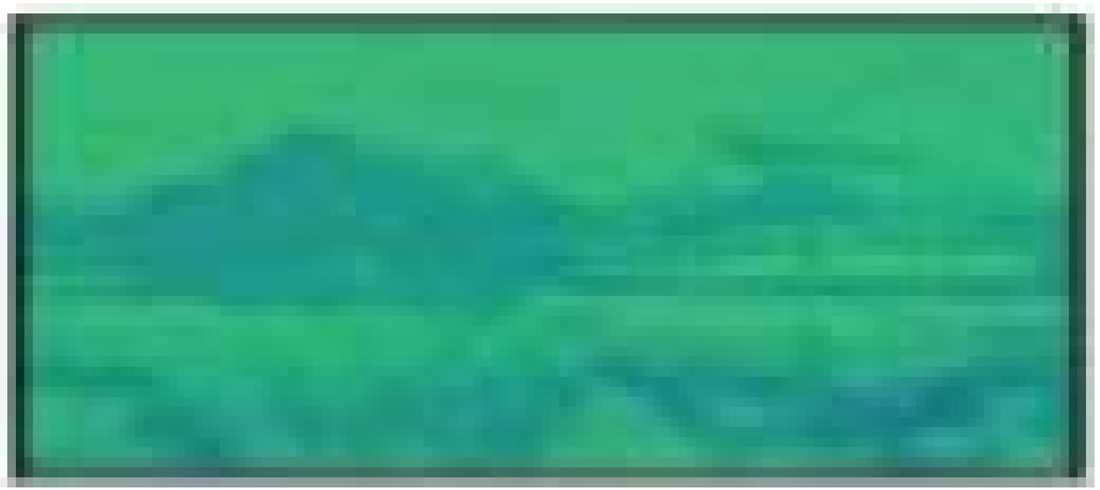


Codes & Layers

[초기 선택 레이어]

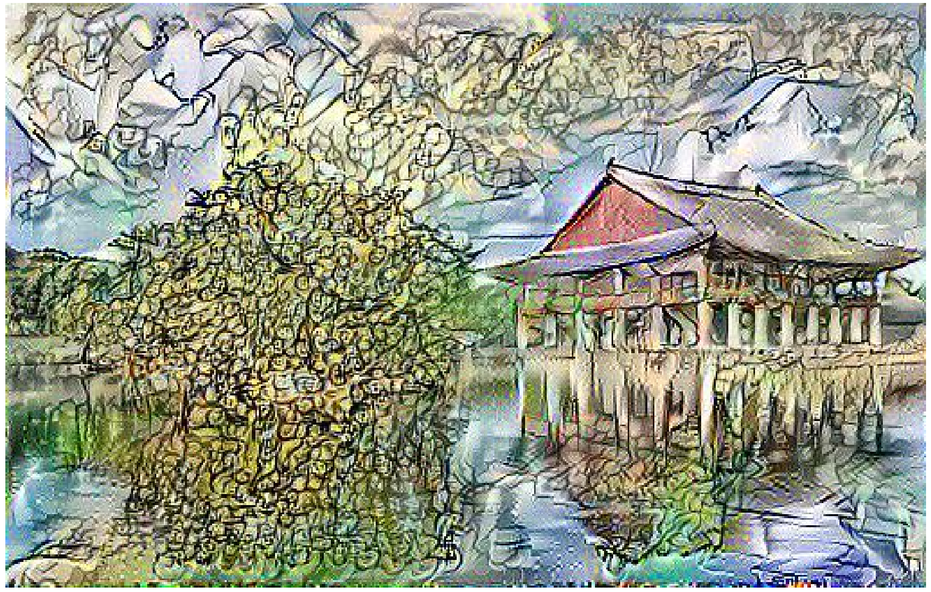


[더 얕은 콘텐츠 레이어]

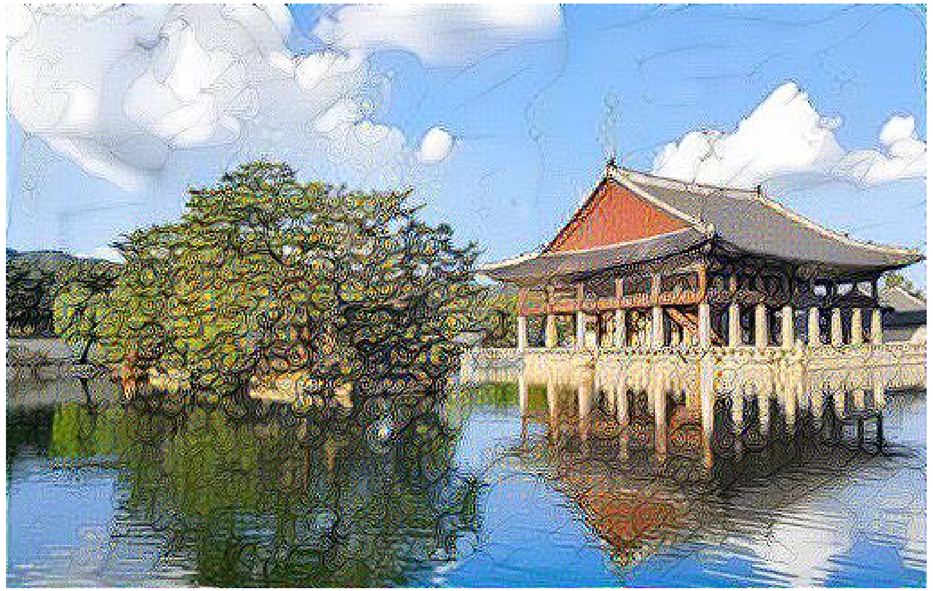


Codes & Layers

[초기 선택 레이어]



[더 얕은 콘텐츠 레이어]



Codes & Layers

```
style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']
```

```
style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1']
```

```
style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1']
```

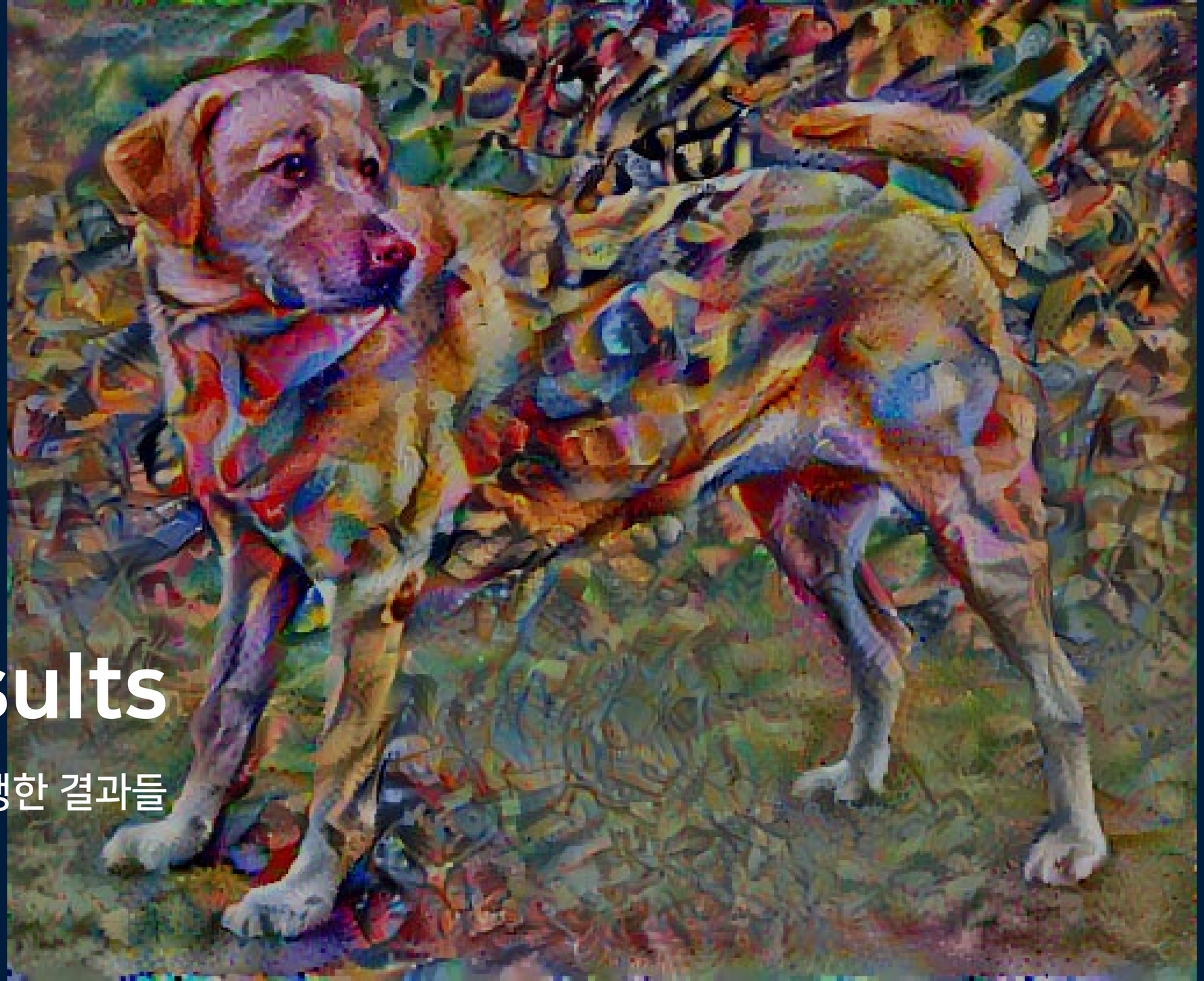
Codes & Layers

스타일 입력값이 줄어들수록 스타일 요소 감소

```
style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1',
                'block5_conv1']
```

```
style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1',
                'block4_conv1']
```

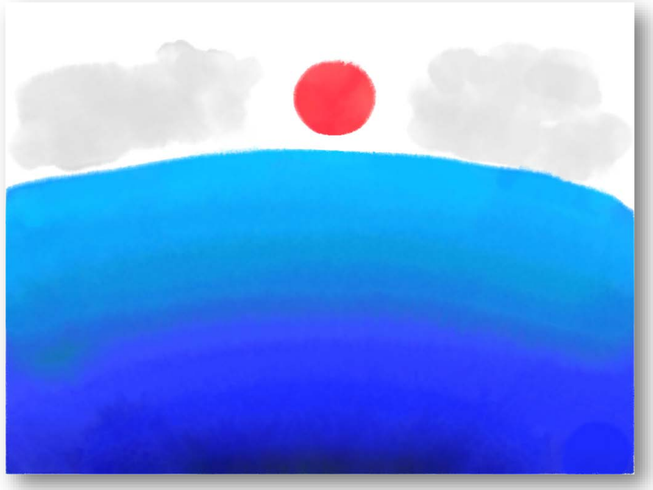
```
style_layers = ['block1_conv1',
                'block2_conv1',
                'block3_conv1']
```

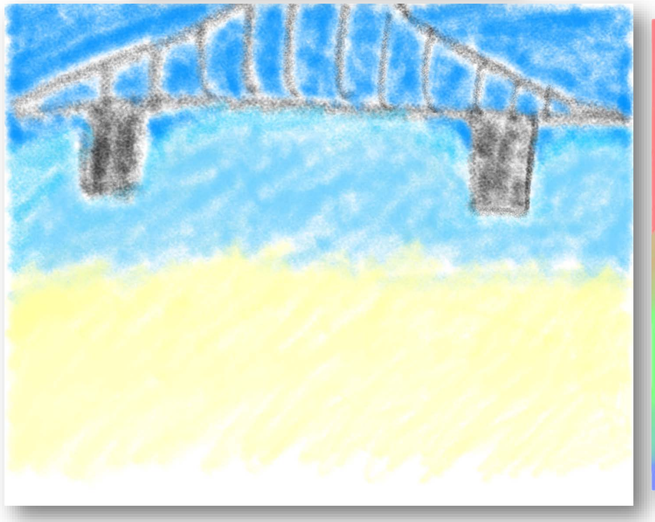


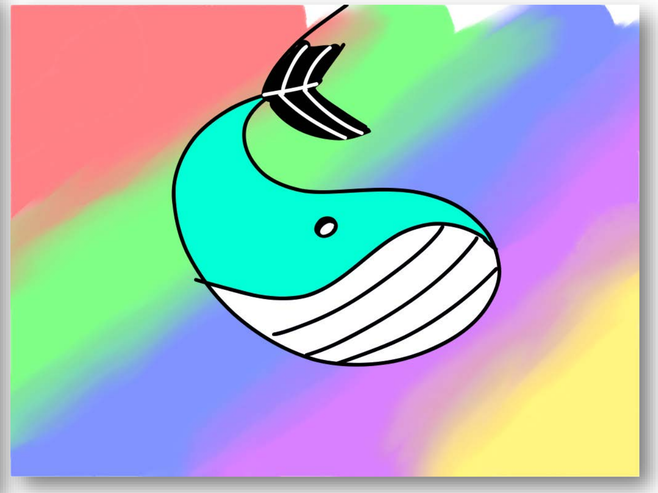
Results

직접 실행한 결과들

Results







Results

Conclusion

앞으로 해볼 것 & 마무리

Conclusion

[원하는 이미지 정보]

[우리들의 화풍]

[결과]

THANK YOU