

Style Transfer

IT 융합공학부 윤세영



유튜브 주소: <https://youtu.be/4R3liWnejbs>

깃허브 주소: <https://github.com/nkolkin13/STROTSS>

목차

기술 내용 설명

코드 분석

수행 결과

문제 발생 및 해결

결과 분석 및 논의, 활용방안

기술 내용 설명

Style Transfer by Relaxed Optimal Transport and Self-Similarity

Style Transfer



이미지의 '내용'은 그대로 유지하면서
이미지의 '스타일'만 변경하는 기법

기술 내용 설명

Style Transfer by Relaxed Optimal Transport and Self-Similarity

STROTSS

objective function. We find that optimizing the Laplacian pyramid, rather than pixels directly, dramatically speeds up convergence. At each scale we make 200 updates using RMSprop, and use a learning rate of 0.002 for all scales except the last, where we reduce it to 0.001.

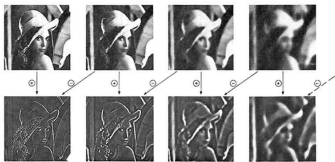


Fig. 5. First four levels of the Gaussian and Laplacian pyramid Gaussian images, upper row, were obtained by applying Gaussian blurring. Each level of the Laplacian pyramid is the difference between the corresponding and next Gaussian pyramid.

Laplacian pyramid

```
if use_pyr:
    s_pyr = dec_lap_pyr(stylized_im, 5)
    s_pyr = [Variable(li.data, requires_grad=True) for li in s_pyr]
else:
    s_pyr = [Variable(stylized_im.data, requires_grad=True)]

optimizer = optim.RMSprop(s_pyr, lr=lr)
```



Fig. 8. Laplacian pyramid code applied to progressive image transmission. High levels of the pyramid are transmitted first to give the receiver a quick but very coarse rendition of the image. The receiver's image is then progressively refined by adding successively lower pyramid levels as these are transmitted. In the example shown here, the leftmost figure shows reconstruction using pyramid levels 4, 3, or just 0.03 bits/pixel. The following four figures show the reconstruction after pyramid levels 3, 2, 1, and 0 have been added. The cumulative data rates are shown under each figure. See companion paper.

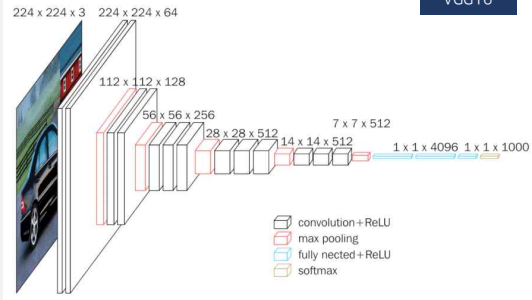
'The Laplacian Pyramid as a Compact Image Code' (1983)

기술 내용 설명

Style Transfer by Relaxed Optimal Transport and Self-Similarity

STROTSS

VGG16



코드 분석

소스코드(깃허브): <https://github.com/nkolkin13/STROTSS>

STROTSS

vgg_pt.py

styleTransfer.py

st_helper.py

contextual_loss.py

utils.py

stylize_objectives.py

pyr_lab.py

코드 분석

소스코드(깃허브): <https://github.com/nkolkin13/STROTSS>

vgg_pt.py

```
class Vgg16_pt(torch.nn.Module):
```

```
    def __init__(self, requires_grad=False):...
```

```
    def forward_base(self, X, rand):...
```

```
    def forward(self, X, inds=[1,3,5,8,11], rand=True):...
```

```
    def forward_cat(self, X, r, inds=[1,3,5,8,11], rand=True, samps=100, forward_func=None):...
```

```
    def forward_diff(self, X, inds=[1,3,5,8,11], rand=True):...
```

VGG16 모델의
특정 레이어까지의
출력을 반환함

```
def __init__(self, requires_grad=False):
```

```
    super(Vgg16_pt, self).__init__()
```

```
    vgg_pretrained_features = models.vgg16(pretrained=True).features
```

사전 훈련된 모델을 가져옴

코드 분석

소스코드(깃허브): <https://github.com/nkolkin13/STROTSS>

styleTransfer.py

Style Transfer를 수행하는 실질적인 부분

```
def run_st(content_path, style_path, content_weight, max_scl, coords, use_guidance, regions, output_path='./output.png'): ...  
  
if __name__ == '__main__': ...
```

입력 인자를 받아 'run_st'를 실행하는 메인 부분

코드 분석

소스코드(깃허브): <https://github.com/nkolkin13/STROTSS>

st_helper.py

논문 저자가 제안하는 Style Transfer 알고리즘이 담겨있음

```
def style_transfer(stylized_im, content_im, style_path, output_path, scl,  
                  long_side, mask, content_weight=0., use_guidance=False, regions=0, coords=0, lr=2e-3):
```

```
    if use_pyr:  
        s_pyr = dec_lap_pyr(stylized_im, 5)  
        s_pyr = [Variable(li.data, requires_grad=True) for li in s_pyr]  
    else:  
        s_pyr = [Variable(stylized_im.data, requires_grad=True)]  
  
    optimizer = optim.RMSprop(s_pyr, lr=lr)
```

수행 결과

content weight == 1.0



수행 결과

content weight == 0.5



수행 결과

content weight == 1.0



문제 발생 및 해결

Import Error

[ImportError: attempted relative import with no known parent package]

기존 코드에서 상대경로를 이용하여 import 했으므로,
실행하는 환경에 맞게 수정해 주는 것이 중요함.



```
from .st_helper import *  
from . import utils  
from .utils import *
```

styleTransfer.py

```
from st_helper import *  
import utils  
from utils import *
```

st_helper.py

```
import utils  
from utils import *  
from vgg_pt import *  
from pyr_lap import *  
from stylize_objectives import objective_class
```

contextual_loss.py

```
import utils
```

stylize_objectives.py

```
from contextual_loss import *  
import utils
```

문제 발생 및 해결

Assertion Error

[AssertionError: Torch not compiled with CUDA enabled]
CPU 전용 노트북에서 실행하였으므로
CUDA와 관련된 코드를 일부 수정함.

```
File ~/ProgramData/Anaconda3/envs/python_venv/lib/python3.7/site-packages/torch/tensor.py:117:
  raise AssertionError("Torch not compiled with CUDA enabled")
AssertionError: Torch not compiled with CUDA enabled
```

stylize_objectives.py

```
#self.z_dist = torch.zeros(1).cuda()
self.z_dist = torch.zeros(1)

#xx = torch.from_numpy(xx).cuda().view(1,1,x_st.size(2),1).float()
#yy = torch.from_numpy(xy).cuda().view(1,1,x_st.size(2),1).float()

xx = torch.from_numpy(xx).view(1, 1, x_st.size(2), 1).float()
yy = torch.from_numpy(xy).view(1, 1, x_st.size(2), 1).float()
```

utils.py

```
# def to_device(tensor):
#     if use_gpu:
#         return tensor.cuda()
#     else:
#         return tensor

def to_device(tensor):
    return tensor
```

문제 발생 및 해결

Imageio Error

Imageio 버전 오류로 이미지 처리에 문제가 발생함.
Import imageio.v3 as iio 를 추가한 뒤,
iio.imread로 수정하여 이미지를 정상적으로 로드함.

```
[[iio.imread(content_path)
```

styleTransfer.py

```
else:
    try:
        regions = [[iio.imread(content_path)[:,:0]*0.+1.], [iio.imread(style_path)[:,:0]*0.+1.]]
    except:
        regions = [[iio.imread(content_path)[:,:]*0.+1.], [iio.imread(style_path)[:,:]*0.+1.]]
```

utils.py

```
def load_path_for_pytorch(path, max_side=1000, force_scale=False, verbose=True):
    com_f = max
    x = iio.imread(path)

    im_a = iio.imread(content_path)
    im_b = iio.imread(style_path)
    s_regions = iio.imread(style_path).transpose(1,0,2)
    c_regions = iio.imread(content_path).transpose(1,0,2)
```

결과 분석 및 논의, 활용방안

결과 분석 및 논의



화질 저하?



디테일 손실?

활용방안

- 딥러닝 기반 스타일 변환 기법을 활용한 인공 달 지형 영상 데이터 생성 방안에 관한 연구 (2022)
- 인공지능 스타일 전이를 활용한 발레 <지젤>에 대한 공연예술의 시각화 연구 (2021)
- 스타일 트랜스퍼를 활용한 평면도 해석 프로세스의 범용성 강화 (2019)
- Style Transfer를 이용한 주행 중인 이동체에서의 야간 차량 인식을 향상 방식 (2019)

