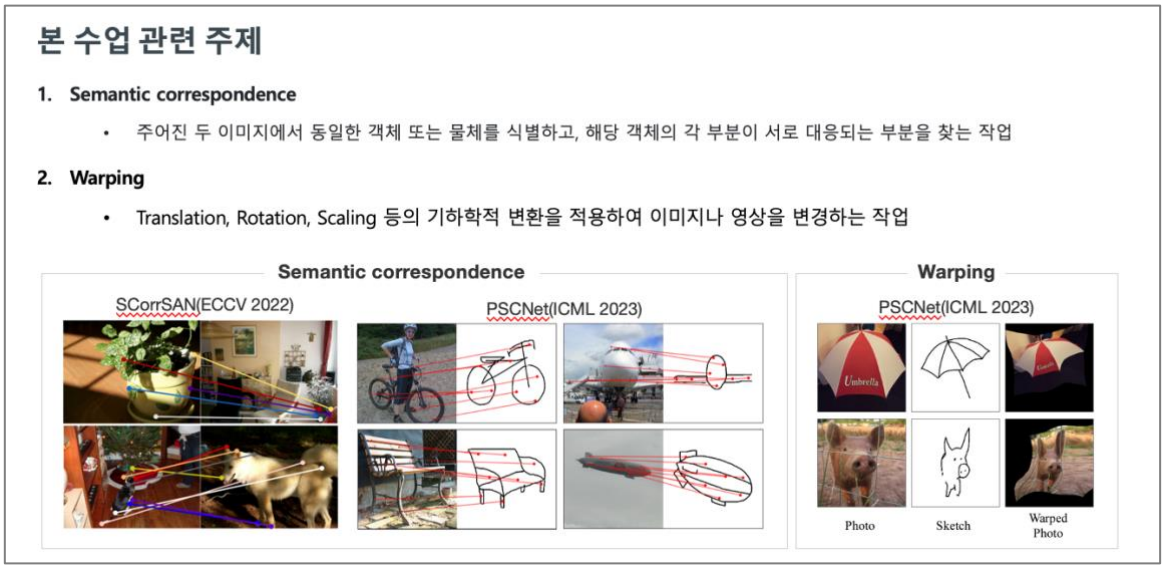
Title: (영상기하학 기말 프로젝트) Style Transfer to Novel Views using Correspondences between Photos and Sketches

Author: 김소정

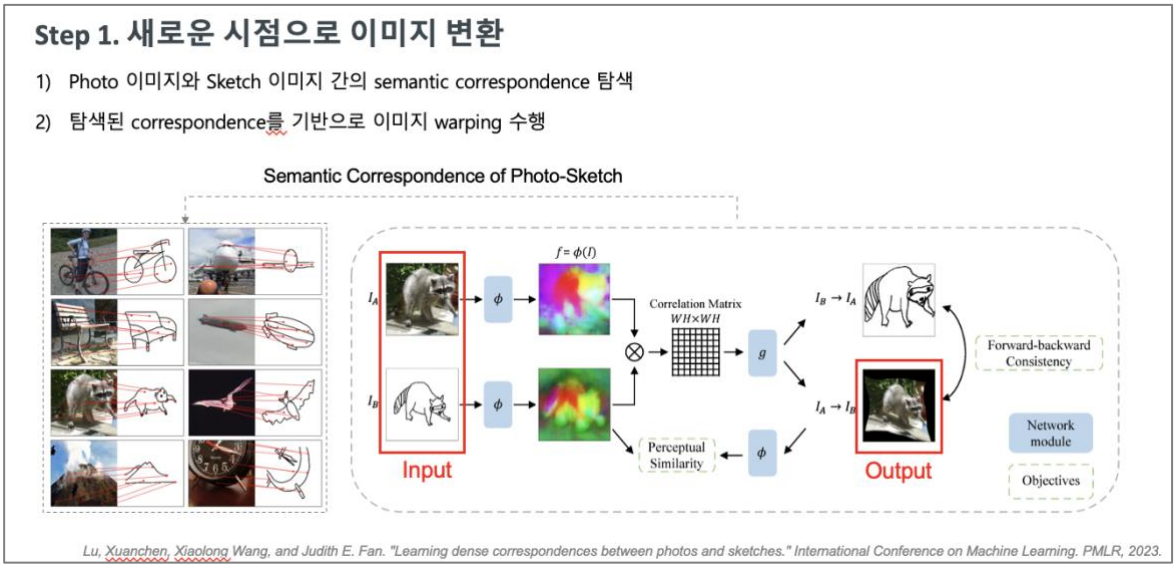
Teaser Figure (Overall concept, target problem, etc.)

- Figure 1: Overall concept

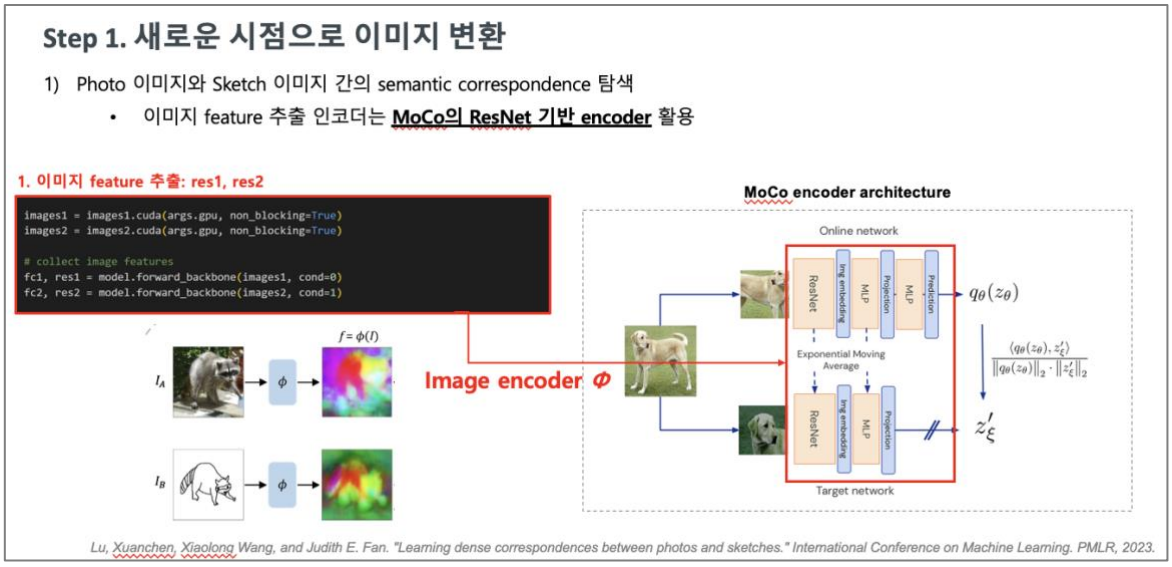


Framework Figure (Detailed architecture, training methods, etc.)

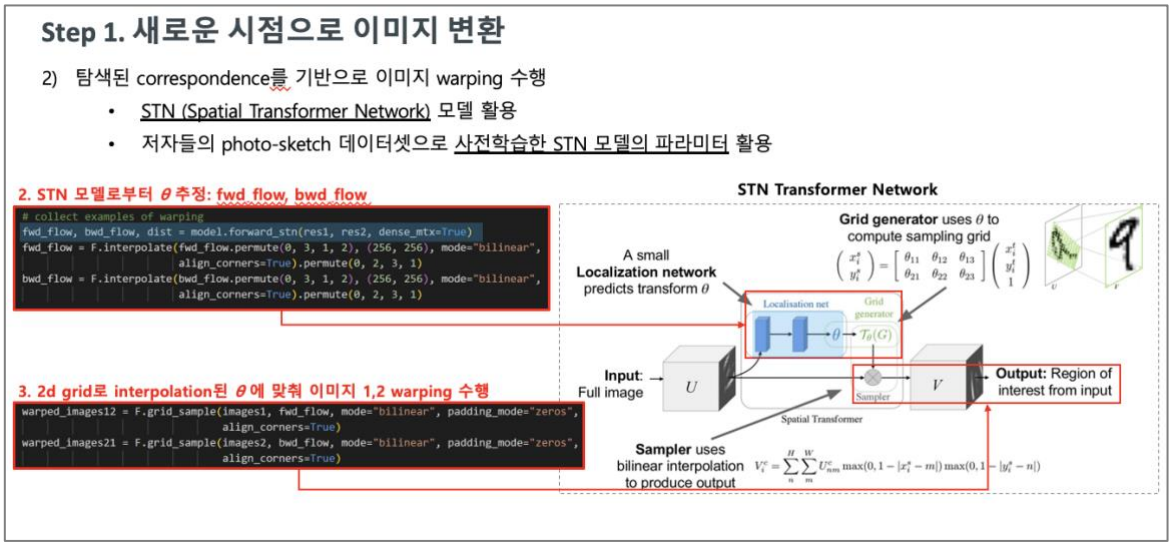
- Figure 2: Detailed architecture-1



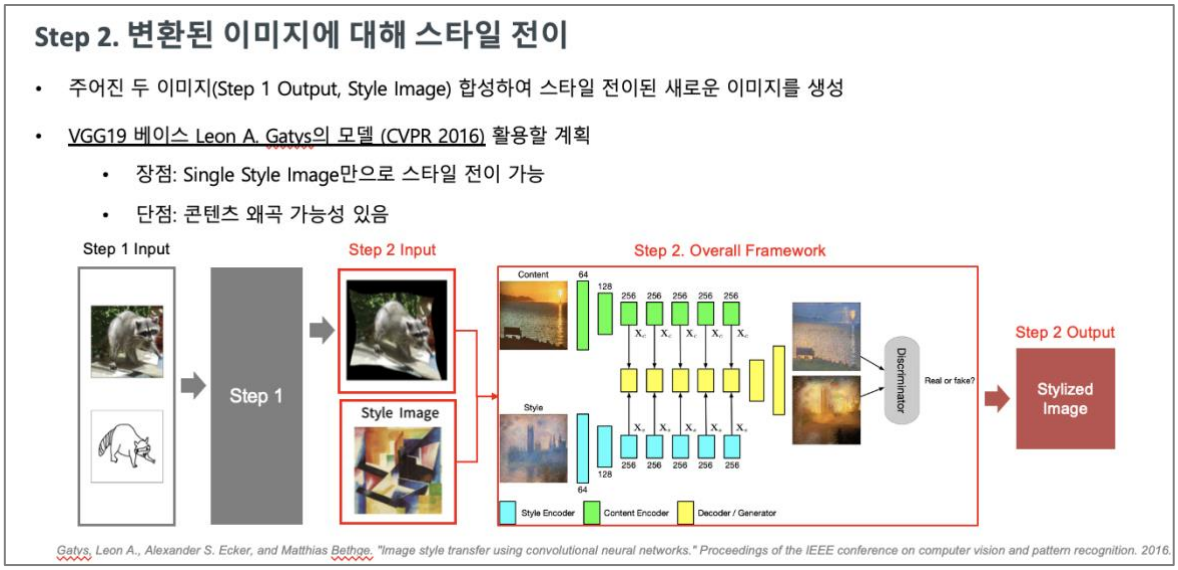
- Figure 3: Detailed architecture-2



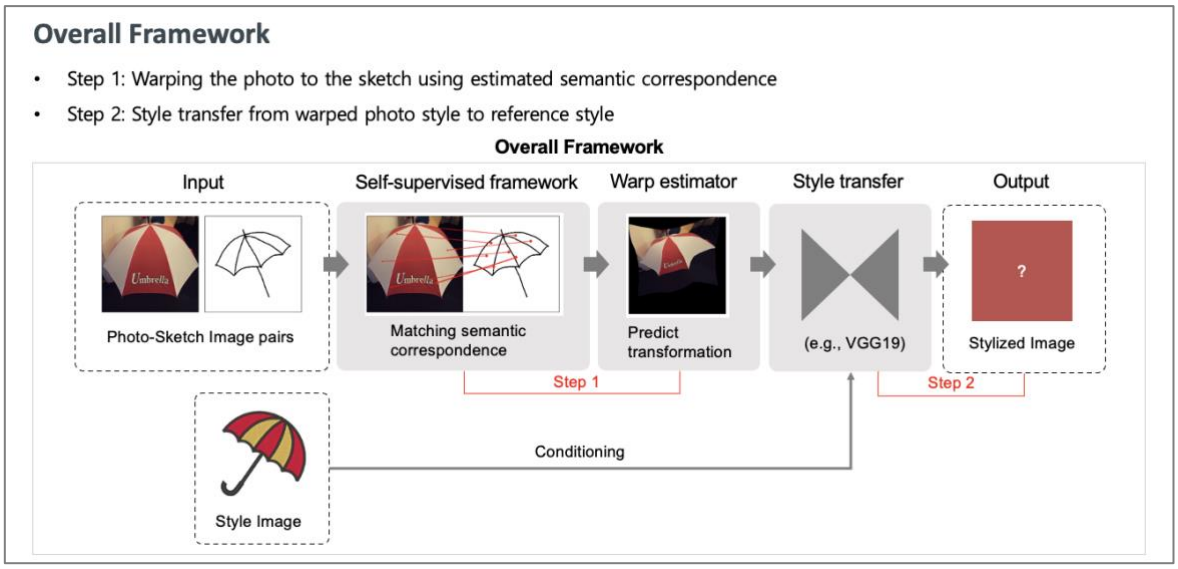
- Figure 4: Detailed architecture-3



- Figure 5: Detailed architecture-4



- Figure 6: Overall architecture



Experiment Figure (Qualitative, quantitative, ablation tests, analysis, etc.)

- Figure 7: Experimental settings-Dataset

1) 데이터셋

- **Photo-Sketchy database**
  - Large-scale collection of sketch-photo pairs
  - The Sketchy database contains 75,471 sketches of 12,500 objects spanning 125 categories
- **MSCOCO 2014**
  - Large-scale object detection, segmentation, key-point detection, and captioning dataset
  - 164K images split into training (83K), validation (41K) and test (41K) sets
- **Wikiart**
  - painting from 195 different artists
  - The dataset has 42129 images for training and 10628 images for testing

- Figure 8: Experimental settings-Metric

2) 평가 지표

1. Metric for semantic correspondence

- **Percentage of correct keypoints (PCK) [3]**
  - 예측된 점과 정답 점들 간의 유클리드 거리 계산

$$\mathrm{PCK}(\mathcal{K}) = \frac{1}{M}\sum_{m=1}^{M}\mathbb{1}[\|k_{\mathrm{GT}}(m) - k_{\mathrm{est}}(m)\| \le \alpha_k \cdot \max(H, W)],$$

- k_{GT}: ground-truth keypoint
- k_{est}: estimated keypoint
- M: number of keypoint pairs
- H, W: width and height of the entire image or object bounding box
- ak: factor.

2. Metric for Warping

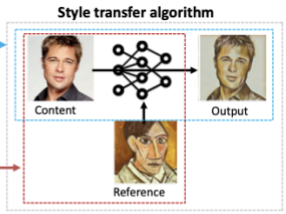
- **KNN accuracy**

3. Metric for Style transfer: Content similarity

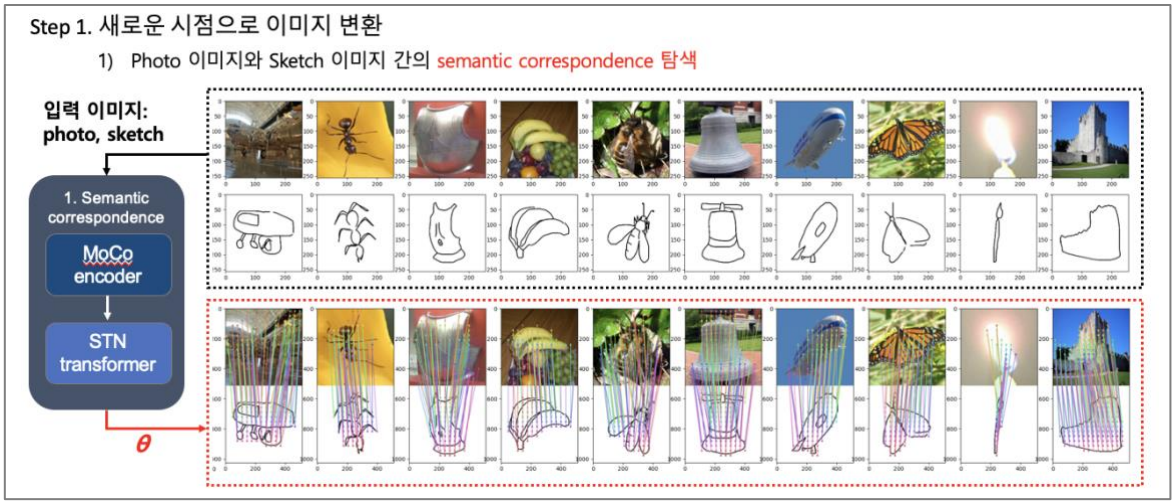
- **VGG Content Loss**
  - For semantic information preservation of content image
  - 콘텐츠 이미지와의 semantic 유사도 계산

4. Metric for Style transfer: Style similarity

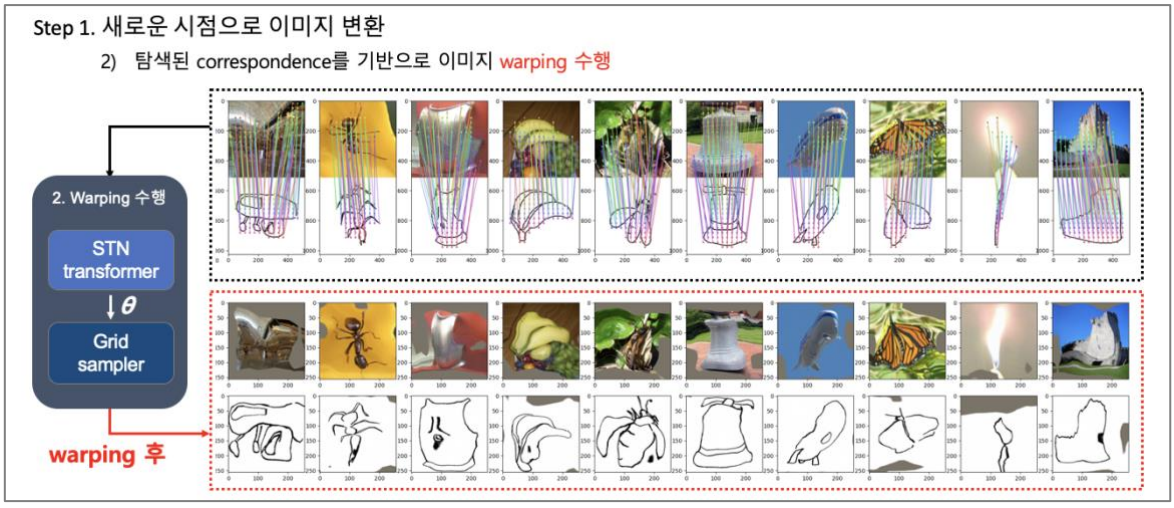
- **VGG Style Loss (Gram Loss)**
  - For style transfer performance of style image(reference image)
  - 레퍼런스 스타일 이미지와의 스타일(질감. 색감 등) 유사도 계산
    - Gram matrix = $E_l = \frac{1}{4N_l^2M_l^2}\sum_{i,j}(G_{ij}^l - A_{ij}^l)^2$
    - Gram Loss = $\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$



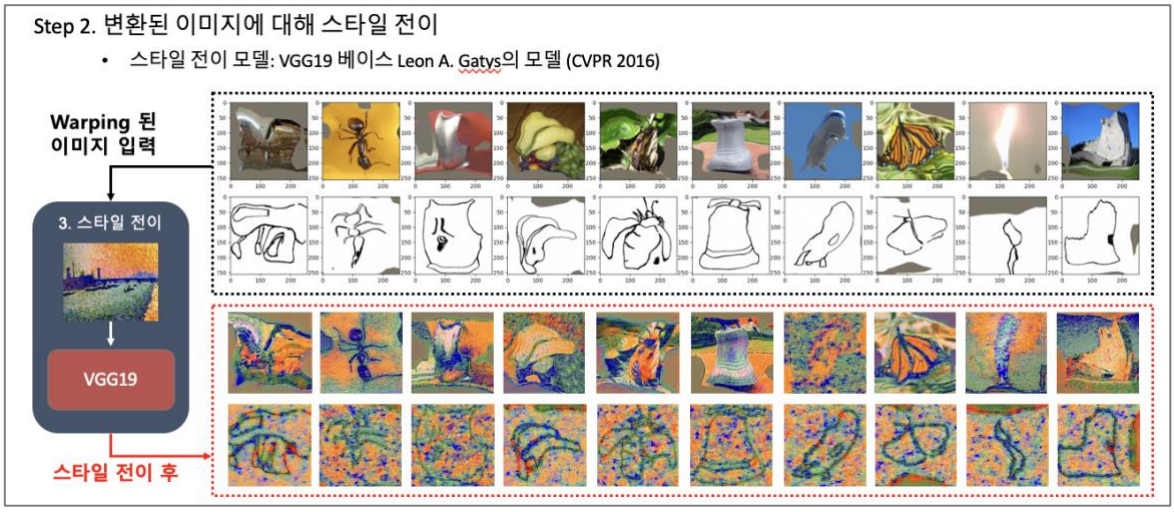
- Figure 9: Experimental Qualitative Results-1



- Figure 10: Experimental Qualitative Results-2



- Figure 11: Experimental Qualitative Results-3



o Figure 12: Experimental Quantitative Results-1

- **Metric for Semantic correspondence: Percentage of correct keypoints (PCK) [3]**
  - 예측된 점과 정답 점들 간의 유클리드 거리 계산

논문 내 실험 성능표

Methods	Encoder	Transfer PCK-5	Transfer PCK-10	Retrain PCK-5	Retrain PCK-10
CNNGeo (Rocco et al., 2018a)	ResNet-101	27.59	57.71	19.19	42.57
WeakAlign (Rocco et al., 2018a)	ResNet-101	35.65	68.76	43.55	78.60
NC-Net (Rocco et al., 2018b)	ResNet-101	40.60	63.50	–	–
DCCNet (Huang et al., 2019)	ResNet-101	42.43	66.53	–	–
PMD (Li et al., 2021)	VGG-16	35.77	71.24	–	–
WarpC-SemanticGLUNet (Truong et al., 2021)	VGG-16	48.79	71.43	56.78	79.70
Ours	ResNet-18	–	–	56.38	**83.22**
Ours	ResNet-101	–	–	**58.00**	**84.93**

Table 1: State-of-the-art comparison for photo-sketch correspondence learning.

추가 실험 후 성능표

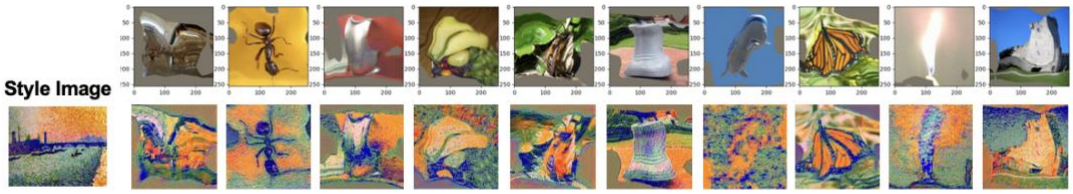
Encoder	총 학습 Epoch	Retrain PCK-5	Retrain PCK-10
ResNet-18	2,503	0.5134	0.8090
	~~2,511~~	0.5416	0.8230

비슷한 결과값 확인

o Figure 13: Experimental Quantitative Results-2

- **Metric for Style transfer: VGG Style Loss (Gram Loss), VGG Content Loss**

Content Image (Top) / Stylized Image (Down)

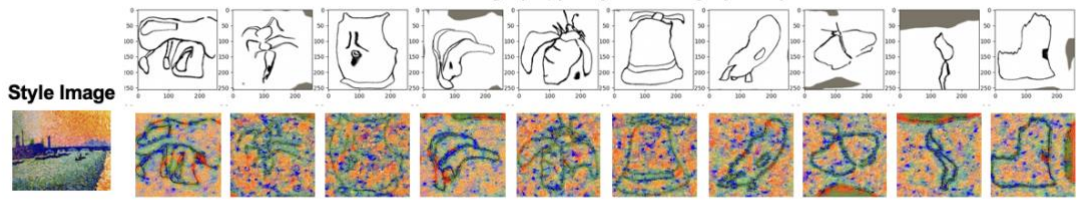


Style Loss	11.0101	4.9306	6.8975	12.6229	15.7456	3.1719	10.7481	4.1821	8.5934	5.4172
Content Loss	40.0513	46.8052	46.6789	44.7521	40.7717	33.4107	49.3909	42.8453	48.4905	42.5540

o Figure 14: Experimental Quantitative Results-3

- **Metric for Style transfer: VGG Style Loss (Gram Loss), VGG Content Loss**

Content Image (Top) / Stylized Image (Down)



Style Loss	15.1441	11.2287	12.9199	18.2418	12.2414	12.0534	13.6700	13.4650	15.4816	15.6202
Content Loss	49.5030	51.5337	52.2461	48.8809	52.5441	52.6480	51.4194	51.0385	49.6445	49.7596

- Figure 15: Experimental Quantitative & Qualitative Results

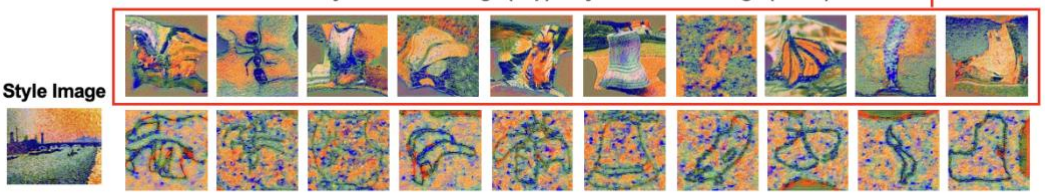
성능 평가

Photo 이미지에서 전반적인 스타일 전이 성능이 더 좋음

- **Metric for Style transfer: VGG Style Loss (Gram Loss), VGG Content Loss**

Stylized Photo Image (Top) / Stylized Sketch Image (Down)

Style Image



Stylized Photo Image (Top)	Style Loss	11.0101	4.9306	6.8975	12.6229	15.7456	3.1719	10.7481	4.1821	8.5934	5.4172
	Content Loss	40.0513	46.8052	46.6789	44.7521	40.7717	33.4107	49.3909	42.8453	48.4905	42.5540
Stylized Photo Image (Down)	Style Loss	15.1441	11.2287	12.9199	18.2418	12.2414	12.0534	13.6700	13.4650	15.4816	15.6202
	Content Loss	49.5030	51.5337	52.2461	48.8809	52.5441	52.6480	51.4194	51.0385	49.6445	49.7596

- Figure 16: Ablation tests - Train Loss with Epipolar constraint

Fundametal matrix F 추정 코드

```
def estimate_fundamental_matrix(self, inputs, targets):
    N, H, W, _ = inputs.shape
    inputs_reshaped = inputs.reshape(-1, 2)
    targets_reshaped = targets.reshape(-1, 2)

    inputs_np = inputs_reshaped.detach().cpu().numpy()
    targets_np = targets_reshaped.detach().cpu().numpy()

    F, mask = cv2.findFundamentalMat(inputs_np, targets_np, cv2.FM_RANSAC)

    if F is None:
        # Return an identity matrix if F is not found
        F = np.eye(3)

    return torch.tensor(F, dtype=torch.float32, device=inputs.device)
```

기존 PSCNet의 Loss function 코드

```
    mse = mse.flatten()
    mse_loss = mse[mask].mean()

    # Estimate Fundamental Matrix F
    F_matrix = self.estimate_fundamental_matrix(inputs, targets)

    # Convert points to homogeneous coordinates
    inputs_reshaped = inputs.reshape(-1, 2)
    targets_reshaped = targets.reshape(-1, 2)

    p1_hom = torch.cat([inputs_reshaped, torch.ones(inputs_reshaped.shape[0], 1).to(inputs.device)], dim=1)
    p2_hom = torch.cat([targets_reshaped, torch.ones(targets_reshaped.shape[0], 1).to(targets.device)], dim=1)

    # Compute Epipolar Constraint loss
    p2_F_p1 = torch.einsum('bi,ij,bj->b', p2_hom, F_matrix, p1_hom)
    epi_loss = torch.mean(p2_F_p1 ** 2)

    # Combine losses
    total_loss = mse_loss + epi_loss
    return total_loss
```

- Figure 17: Ablation tests – Quantitative Results-1

- **Metric for Warping: KNN accuracy (number of neighbors=1)**

	Original PSCNet		튜닝 후 PSCNet – v1	
총 학습 Epoch	**sketch2photo**	**photo2sketch**	**sketch2photo**	**photo2sketch**
2501	0.062240	**0.023840**	**0.063840**	0.022720
2502	**0.060640**	**0.023200**	0.056320	0.017280
2503	**0.060640**	**0.023840**	0.057760	0.016640
2504	**0.061120**	**0.023520**	0.058720	0.017280
2505	**0.061600**	**0.022880**	0.058400	0.017920
2506	**0.061120**	**0.023360**	0.057120	0.017280
2507	**0.061120**	**0.023040**	0.057600	0.017920
2508	**0.061920**	**0.023040**	0.051520	0.015840
2509	**0.061600**	**0.023520**	0.049760	0.015520
2510	**0.061120**	**0.022400**	0.050400	0.015200
2511	**0.061440**	**0.023040**	0.050560	0.016160

General description of your target problem (~5 lines)

기술이 발전함에 따라 증강 현실(AR)과 가상 현실(VR)에서는 다양한 시각에서 이미지를 생성하는 필요성이 점점 커지고 있습니다. 또한, 스타일 전이(Style Transfer)를 통해 증강 현실(AR)에서의 콘텐츠를 다양화하고 시각적 효과를 향상시킬 수 있습니다. 따라서, 새로운 시점의 구도로 변환된 이미지를 스타일 전이를 통해 사용자의 취향에 맞게 변환함으로써 시각적 경험을 풍부하게 하고 개별적인 미적 욕구를 충족시키고자 합니다.

The limitation of the target problem that you want to solve (~7 lines)

사용자가 원하는 다양한 시점의 구도를 파악하고 이에 맞춰 이미지를 변환하는 것이 어렵습니다. 사용자가 원하는 시점의 구도를 정확히 반영하지 못하면, 사용자 맞춤형 시각적 콘텐츠 제공의 가능성이 저해됩니다. 또한, 스타일 전이(Style Transfer)를 통해 생성된 이미지가 사용자의 개별적인 취향을 완전히 반영하지 못하거나, 자연스럽지 않은 결과물이 나타날 수 있습니다. 이러한 문제는 사용자 만족도를 감소시키며, 특히 실시간 상호작용이 중요한 AR 환경에서는 큰 제약이 됩니다.

The solution you've found to solve the limitation (~7 lines)

스케치는 사용자가 직관적으로 원하는 새로운 시점을 표현할 수 있는 수단을 제공하기 때문에, 이미지의 구도를 변환하는 데 유용합니다. 따라서 스케치 이미지를 사용하면 사용자가 원하는 새로운 시점을 쉽고 간편하게 표현할 수 있습니다. 이를 위해 스케치와 원본 이미지 간의 semantic correspondence를 찾아 warping을 수행함으로써, 스케치를 기반으로 이미지를 다양한 시각으로 변환할 수 있습니다. 또한, 새로운 시점으로 변환된 이미지를 사용자의 취향에 맞게 스타일화하기 위해 스타일 전이(Style Transfer)를 활용합니다. 이 과정을 통해 변환된 이미지는 사용자에게 더욱 매력적이고 맞춤화된 시각적 경험을 제공합니다.

The detailed description of your framework (~ 100 lines)

본 프로젝트는 PSCNet (2023 PMLR), VGG (2016 CVPR) 모델을 기반으로 하여, 다양한 구도로 이미지를 변환하고 스타일 전이를 수행하는 프레임워크를 구현합니다. 전체적인 프레임워크는 [Figure 6]과 같습니다.

- Step 1: 이미지 구도 변환 [Figure 2]
    - 피처 추출 및 매칭: 먼저, 이미지 구도 변환 단계에서는 피처 추출과 매칭 과정을 거칩니

다. [Figure 3]와 같이 ResNet을 기반으로 한 사전학습된 MoCo 인코더를 사용하여 이미지로부터 피처를 추출하고, 추출된 피처를 통해 uvmap 기반의 피처 매칭을 수행합니다. 이를 통해 시맨틱 코레스폰던스를 찾고, 포토 이미지를 스케치 이미지 구도로, 또는 스케치 이미지를 포토 이미지 구도로 변환할 수 있습니다.

  - 와핑 수행: 이후, [Figure 4]과 같이 STN(Spatial Transformer Network) 네트워크를 사용하여 와핑을 수행합니다. STN을 통해 추정된 세타 값을 이용하여 fwd flow와 bwd flow를 얻고, fwd_flow 세타를 활용하여 포토 이미지를 스케치 이미지 구도로 변환하거나, bwd_flow 세타를 활용하여 스케치 이미지를 포토 이미지 구도로 변환합니다.

- Step 2: 스타일 전이 [Figure 5]

  - 스타일 전이: 두 번째 단계인 스타일 전이에서는 2016년에 발표된 VGG19 모델을 사용합니다. [Figure 5]와 같이 앞서 와핑된 이미지와 스타일 이미지를 입력으로 넣어 최종적으로 스타일 전이된 이미지를 얻습니다.

- Overall Framework [Figure 6]

  - [Figure 6]과 같이 전체적인 프레임워크는 구도가 서로 다른 포토 이미지와 스케치 이미지를 입력으로 받아, PSCNet 모델을 통해 와핑된 이미지를 얻고, 이를 스타일 전이 모델에 스타일 이미지와 함께 입력하여 최종 스타일 전이된 이미지를 출력합니다. 이 프레임워크를 통해 다양한 구도로 변환된 이미지를 사용자의 취향에 맞게 스타일 전이하여 시각적 경험을 풍부하게 할 수 있습니다.

Experimental results (~ 100 lines)

- Experimental settings

  - Dataset: 본 실험에서는 크게 세 가지 데이터셋을 활용하였습니다. [Figure 7]과 같이 포토 이미지와 스케치 이미지 쌍과 정답 코레스폰던스가 있는 데이터셋, 콘텐츠 이미지를 위한 MSCOCO 데이터셋, 스타일 이미지를 위한 위키아트 데이터셋을 활용하였습니다.

    - Photo-Sketchy database: Large-scale collection of sketch-photo pairs, The Sketchy database contains 75,471 sketches of 12,500 objects spanning 125 categories
    - MSCOCO 2014: Large-scale object detection, segmentation, key-point detection, and captioning dataset, 164K images split into training (83K), validation (41K) and test (41K) sets
    - Wikiart: painting from 195 different artists, The dataset has 42129 images for training and 10628 images for testing

- Metric: 평가 지표는 [Figure 8]과 같이 총 네 가지를 사용하였습니다. 첫 번째 단계인 시맨틱 코레스폰던스와 와핑을 평가하기 위해 Percentage of Correct Keypoints (PCK)와 KNN 정확도를 사용하였고, 두 번째 단계인 스타일 전이를 평가하기 위해 VGG Content Loss와 VGG Style Loss(Gram Loss)를 활용하였습니다.
    - Metric for semantic correspondence: Percentage of correct keypoints (PCK) [3]
    - Metric for Warping: KNN accuracy
    - Metric for Style transfer - Content similarity: VGG Content Loss
    - Metric for Style transfer - Style similarity: VGG Style Loss (Gram Loss)

- Experimental Results: Quantitative results
    - Step 1 이미지 와핑 결과: PCK 지표를 통해 시맨틱 코레스폰던스의 정확도를 측정하였으며, [Figure 12]의 수치값과 같이 기존 모델과 유사한 성능을 확인할 수 있었습니다. 또한, 와핑 성능은 KNN 정확도를 활용하여 평가하였고, [Figure 17] 표의 2, 3열의 수치값과 같이 오리지널 모델과 유사한 수치값을 얻었습니다.
    - Step 2 스타일 전이 결과: 스타일 전이 성능 평가에서는 포토 이미지와 스케치 이미지에 대해 각각 VGG Content Loss와 VGG Style Loss를 측정하였습니다. 그 결과, [Figure 13,14,15]에서의 수치값과 같이 포토 이미지가 스케치 이미지보다 더 높은 스타일 전이 성능을 보임을 확인할 수 있었습니다.

- Experimental Results: Qualitative results
    - Step 1 이미지 와핑 결과: Step 1의 이미지 와핑 실험 결과, 다양한 이미지에 대해 변환 결과물은 [Figure 9]의 모습과 같습니다. 포토 이미지와 스케치 이미지를 MoCo 인코더와 STN 트랜스포머 모델에 입력하여 세타 값을 구하고, 이를 활용하여 매칭되는 코레스폰던스 점들을 시각화하여 다음과 같은 결과를 얻을 수 있습니다. 이후, 이 세타 값을 이용하여 와핑을 수행하면 [Figure 10]과 같이 와핑된 이미지를 얻을 수 있습니다.
    - Step 2 스타일 전이 결과: 스타일 전이 단계에서는 와핑된 이미지와 스타일 이미지를 VGG19 모델에 입력하여 최종적으로 스타일 전이된 이미지를 얻습니다. 그 결과는 [Figure 11]과 같습니다.

- Experimental Results: Analysis
    - Step 1 이미지 와핑: 스케치 이미지를 통해 사용자가 원하는 시점의 구도를 추정하고 이미지를 변환하고자 하지만, 스케치와 원본 이미지 간의 semantic correspondence을 정확히 찾는 것이 쉽지 않습니다. 본 프로젝트에서는 semantic correspondence을 찾기 위해 사용한 방법은 uv 맵 기반의 feature matching을 사용하였습니다. 그러나 이

방법은 대응하는 점들이 Epipolar Constraint를 성립하지 않을 가능성이 높아, warping 결과 생성된 이미지가 실제 모습보다 왜곡되는 문제가 발생합니다. 따라서 Semantic Correspondence를 취득할 때 Epipolar Constraint를 고려하도록 손실 함수를 추가하여 학습 방법을 개선하고자 Ablation test로 추가 실험을 진행해보았습니다.

- Step 2 스타일 전이: 실험 결과, 스케치 이미지에서 스타일 전이 성능이 포토 이미지에서보다 상당히 낮게 나오는 것을 볼 수 있었습니다. 이러한 현상이 나타난 원인으로, 만약 스타일 이미지와 콘텐츠 이미지가 유사할수록 스타일 전이가 더 잘된다는 가설이 맞다면, 포토 이미지가 스케치보다 색감 등의 이유로 0~255 사이의 다양한 값을 가지게 되어 스타일 이미지와 더 유사하게 보일 수 있고, 이로 인해 포토 이미지에서 스타일 전이가 더 잘 이루어지는 것으로 추측해볼 수 있습니다. 또한, 부가적인 이유로는 모델 성능의 한계를 생각해볼 수 있습니다.

Ablation tests & Analysis (~ 50 lines)

- **Ablation tests:** Epipolar Constraint Loss for Step 1
  - Step 1 이미지 와핑 과정에서의 성능을 개선하기 위해 Epipolar constraint를 고려한 손실 함수를 추가하여 모델 학습을 진행하는 실험을 해보았습니다. 이 실험을 통해 Epipolar Constraint를 반영한 학습 방법이 왜곡 문제를 줄이고 보다 정확한 semantic correspondence를 취득하는 데 도움이 되는지 평가했습니다.
  - 구현 코드: Epipolar Constraint를 고려하도록 손실 함수는 [Figure 16]와 같이 구현하였습니다. estimate_fundamental_matrix 함수를 통해 주어진 inputs와 targets로부터 Fundamental Matrix F를 추정합니다. 이 과정에서 inputs와 targets를 (N, 2) 형식으로 변환한 후, numpy 배열로 변환하고, OpenCV의 findFundamentalMat 함수를 사용하여 F를 추정합니다. 만약 F가 추정되지 않으면 기본값으로 단위 행렬을 설정합니다. forward 메서드는 입력된 inputs와 targets를 이용해 손실을 계산하며, 먼저 MSE 손실을 계산하고 무효값을 무시하는 마스크를 적용하여 마스킹된 MSE 손실의 평균을 계산합니다. 그런 다음 estimate_fundamental_matrix 함수를 호출하여 Fundamental Matrix F를 추정하고, inputs와 targets를 homogeneous coordinates로 변환한 후, F를 사용하여 Epipolar Constraint 손실을 추가로 계산합니다. 최종적으로 Masked MSE 손실과 Epipolar Constraint 손실을 합산하여 총 로스 값을 반환합니다.
  - 실험 결과 및 문제 분석: 실험 결과, [Figure 17]에서 볼 수 있듯이 KNN 정확도에서 epipolar constraint를 고려한 손실 함수를 추가하여 학습한 경우, 에폭이 증가함에 따라 기존 모델보다 정확도가 낮아지는 현상이 발생했습니다. 이러한 결과의 원인 중 하나로 estimate_fundamental_matrix 함수에서의 계산 문제를 생각해볼 수 있습니다.

이 함수는 주어진 입력 데이터로부터 Fundamental Matrix F를 추정하며, F를 추정하지 못할 경우 기본값으로 단위 행렬을 반환하도록 구현되어 있습니다. 이 과정에서 잘못된 F가 사용되면 epipolar constraint가 잘못 적용되어 모델 학습에 부정적인 영향을 미칠 수 있습니다. 특히, 단위 행렬을 사용할 경우 실제 대응 관계를 반영하지 않아 잘못된 제약 조건을 부과하게 되어 성능 저하로 이어질 수 있습니다. 따라서 본 코드를 바탕으로 한 학습 손실을 사용할 경우 성능 저하가 발생한 원인을 estimate_fundamental_matrix 함수에 의한 것으로 생각해 볼 수 있고, 이 문제를 해결하기 위해 F 추정 정확성을 높이고, 단위 행렬 사용을 피하며, 노이즈를 줄이는 등의 개선이 필요하다고 생각됩니다.

- **Ablation analysis:** Style transfer for Step 2
    - [Figure 13, 14, 15]와 같이 스케치 이미지에서 스타일 전이 성능이 포토 이미지에서보다 상당히 낮게 나오는 것을 확인했고, 그 원인 중 하나로, 포토 이미지는 스케치보다 색감 등의 이유로 스타일 이미지와 유사하여 스타일 전이가 더 잘 이루어질 수 있는 반면, 스케치 이미지는 단순한 선으로 구성되어 있어 스타일 이미지와의 유사성이 낮아 성능이 떨어질 수 있다고 추측하였습니다. 또한, 현재 모델의 성능 한계도 한 원인으로 생각됩니다.
    - 따라서, 스케치 이미지에서의 스타일 전이의 성능을 더 높이기 위해 색감이나 깊이와 같은 추가 정보를 컨디셔닝으로 사용하거나, 다른 스타일 전이 모델을 활용할 수 있습니다. 또한, 스케치 이미지와 유사한 데이터를 더 많이 확보하여 모델을 학습시켜 볼 수도 있을 것 같습니다.
    - 또한, 스케치 이미지 뿐만 아니라 포토 이미지를 포함한 전반적인 스타일 전이 성능을 향상시키기 위해 다른 스타일 전이 모델을 사용하거나 학습 과정에 스타일 로스를 도입하여 더 적절한 스타일 피처를 추출할 수 있도록 모델 아키텍처를 개선하는 방법도 고려할 수 있습니다.

Conclusion & Future works (~ 10 lines)

본 프로젝트에서는 스케치 이미지를 통해 새로운 시점의 구도를 표현하고 스타일 전이를 수행하는 주제로 실험을 진행했습니다. 기존 방법으로는 실험 결과 와핑된 이미지가 실제 물체의 통상적인 모양으로부터 상당히 왜곡되는 문제가 있었습니다. 이를 개선하기 위해 epipolar constraint를 고려한 손실 함수를 도입한 추가 실험을 진행했으나, KNN 정확도가 낮아지는 경향을 보였습니다. 이는 Fundamental Matrix 추정의 부정확성 때문으로 추측됩니다. 또한, 스케치 이미지의 스타일 전이 성능이 낮은 이유로 색감 등의 유사성 부족을 추측하였고, 이를 개선하기 위해 추가 정보를 컨디셔닝하거나 다른 스타일 전이 모델을 활용하는 방안을 생각해보았습니다. 추후 프로젝트에서는 이러한 실험 결과를 바탕으로 F 추정 정확성을 높이고, 다양한 스타일 전이 모델과 손실 함수를 적용하여 성능을 향상시켜볼 수 있을 것 같습니다.