Sem-CS: Semantic CLIPStyler for Text-Based Image Style Transfer

Chanda Grover Kamra[1], Indra Deep Mastan[2], Debayan Gupta[1]

[1] Ashoka University, Computer Science, India.

[2] LNM Institute of Information Technology, Jaipur, India.

30th International Conference on Image Processing

ICIP 2023, Kuala Lumpur, Malaysia

Agenda

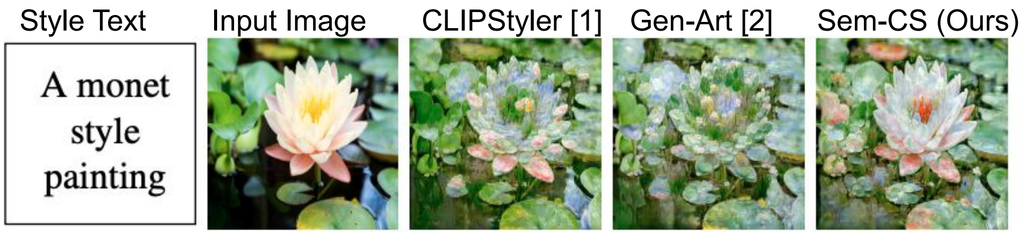
- Motivation
- Method
- Algorithm
- Results
- References

Motivation

- Over Stylization
  - *Distortion of content features.*

Motivation

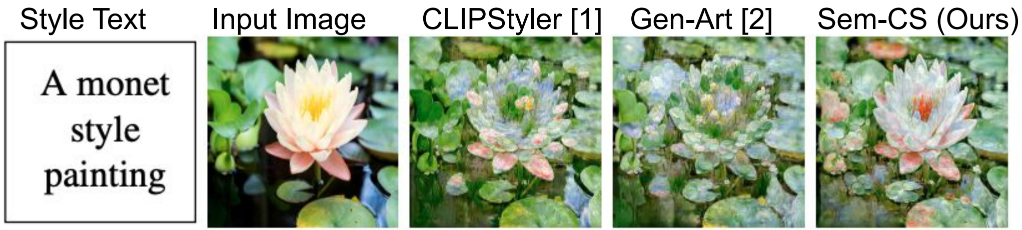
- Over Stylization
    - *Distortion of content features.*



Motivation

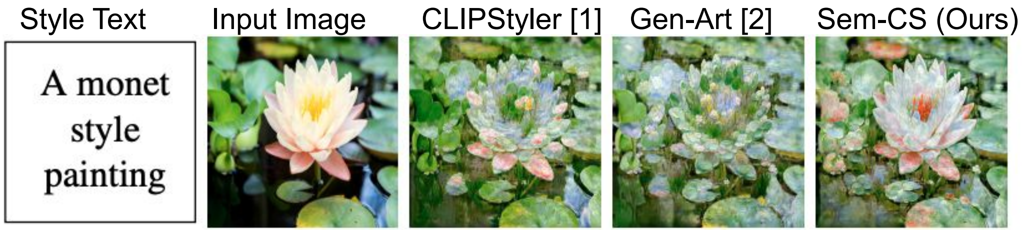
- Over Stylization
    - *Distortion of content features.*

- Content Mismatch
    - *Style Spillover between dissimilar objects.*



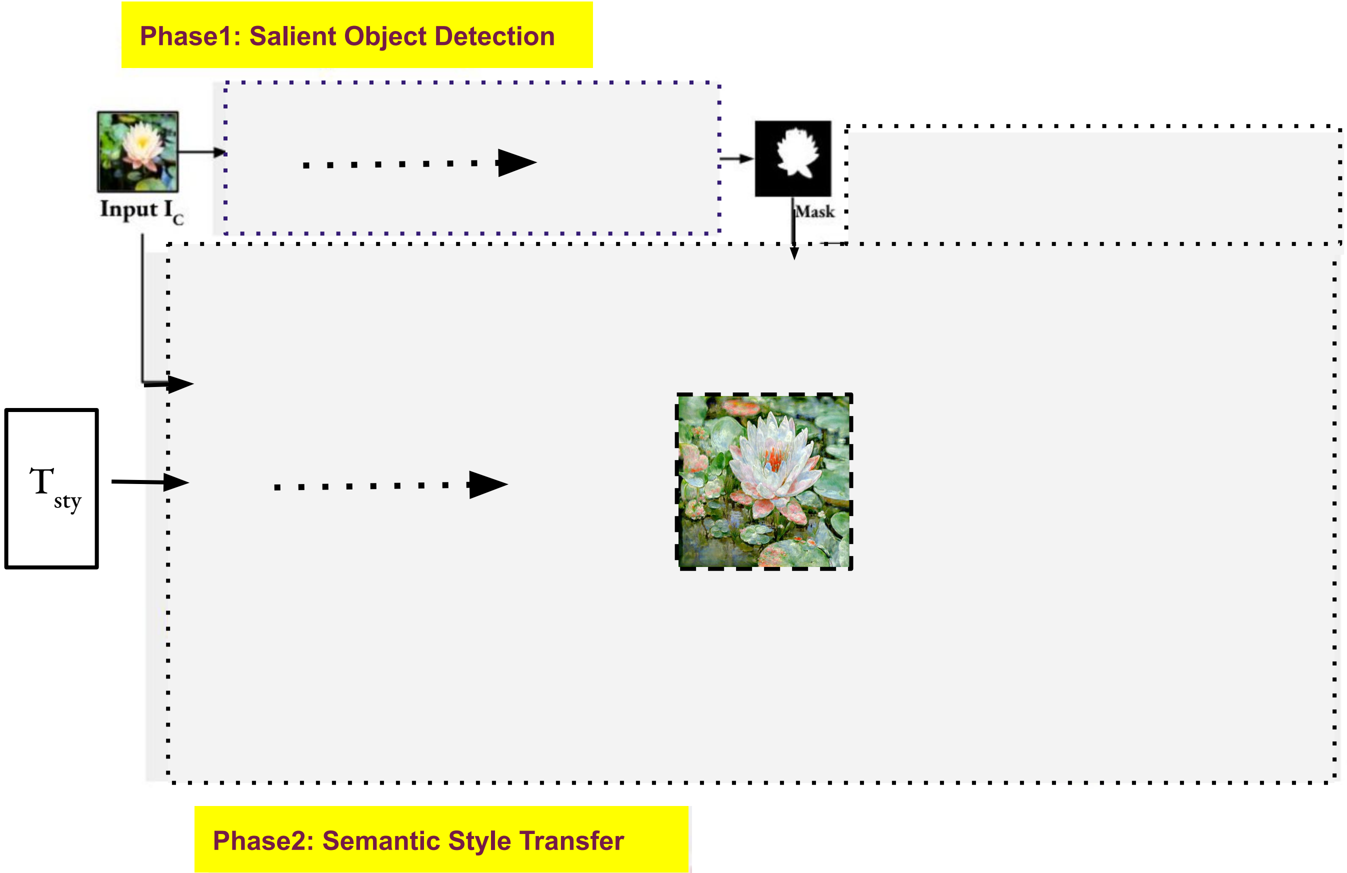
Motivation

- Over Stylization
  - *Distortion of content features.*



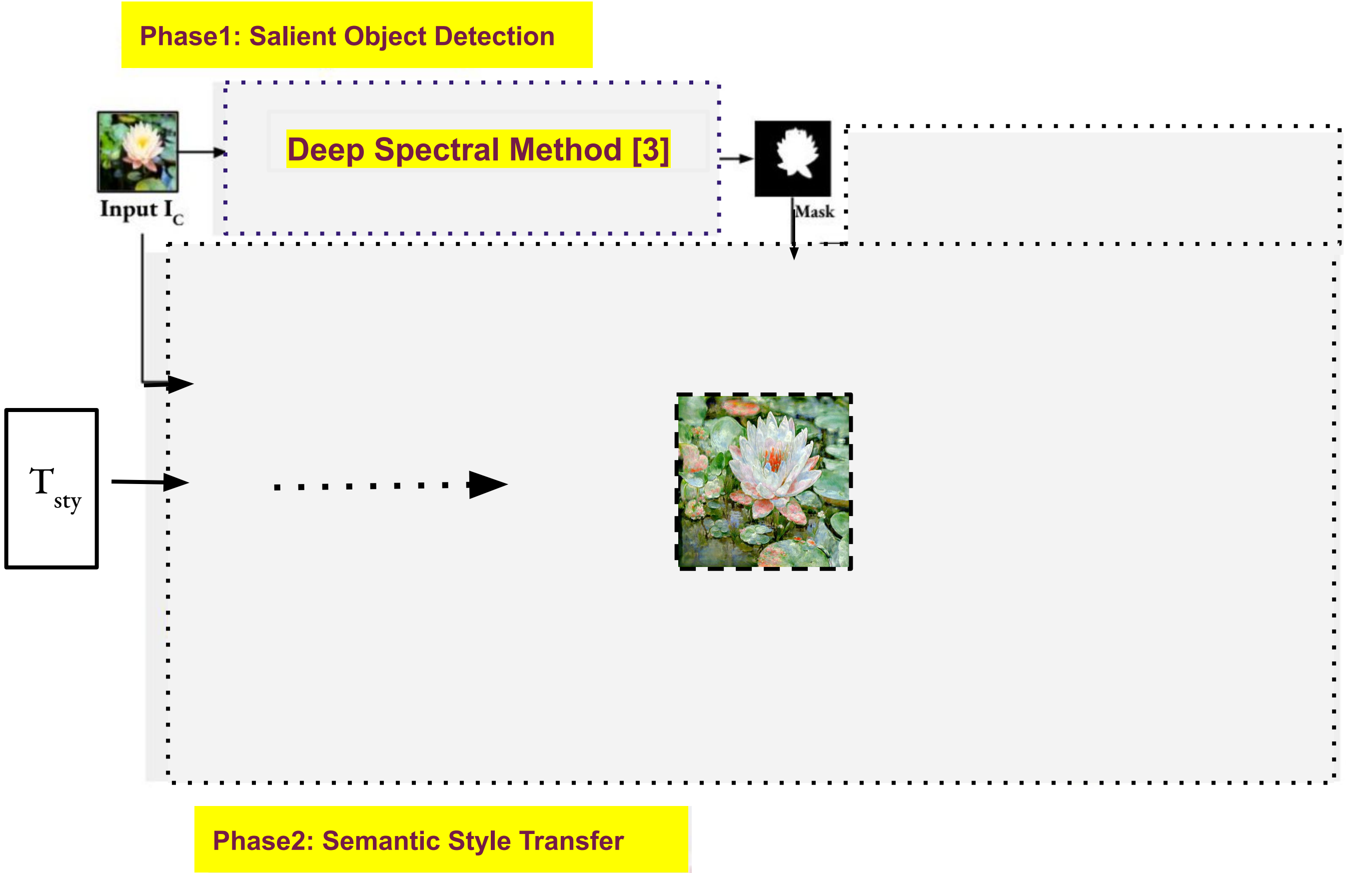
- Content Mismatch
  - *Style Spillover between dissimilar objects.*

Method

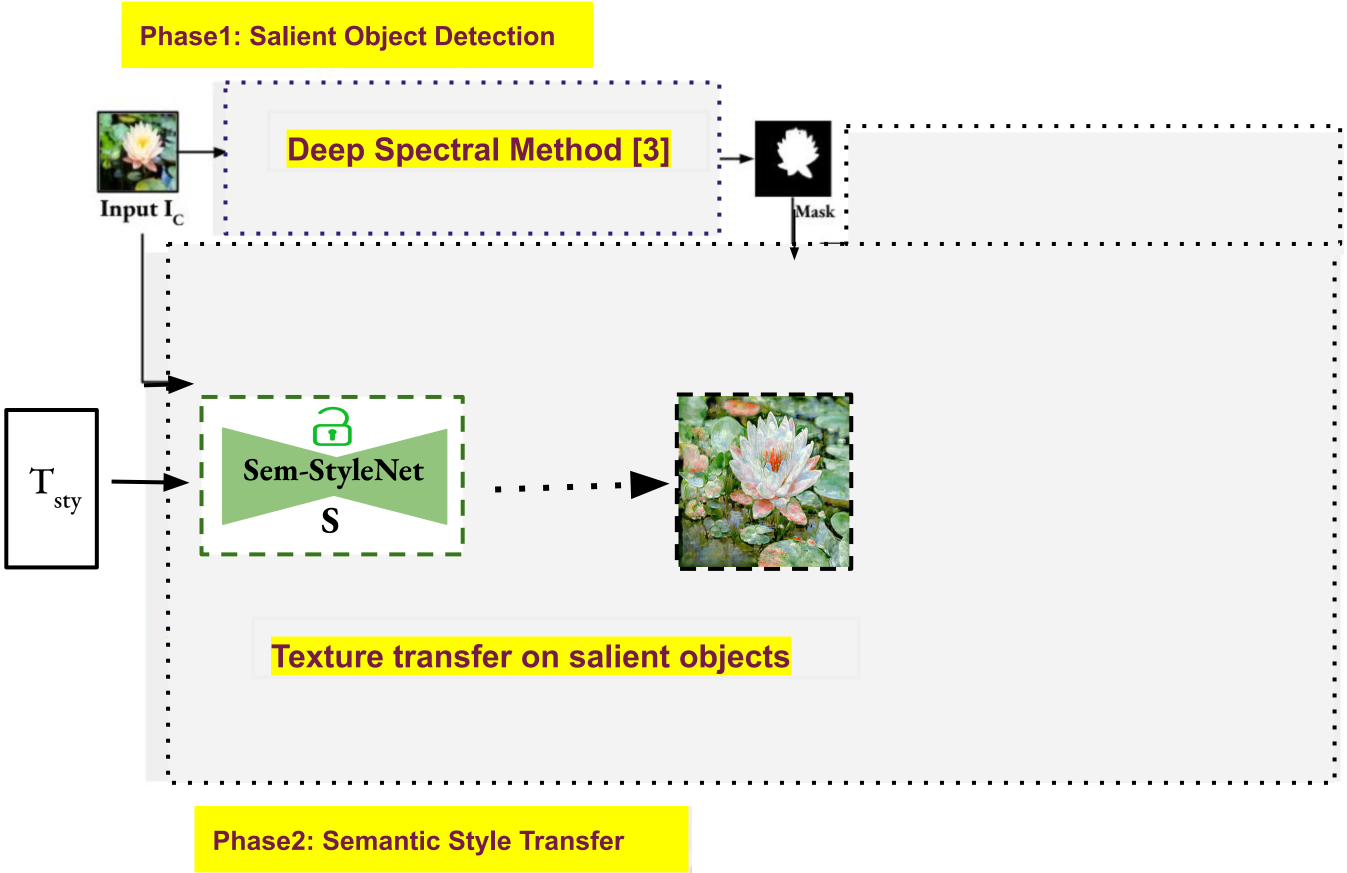


Phase1: Salient Object Detection
Input I_C
Mask
T_{sty}
Phase2: Semantic Style Transfer

Method

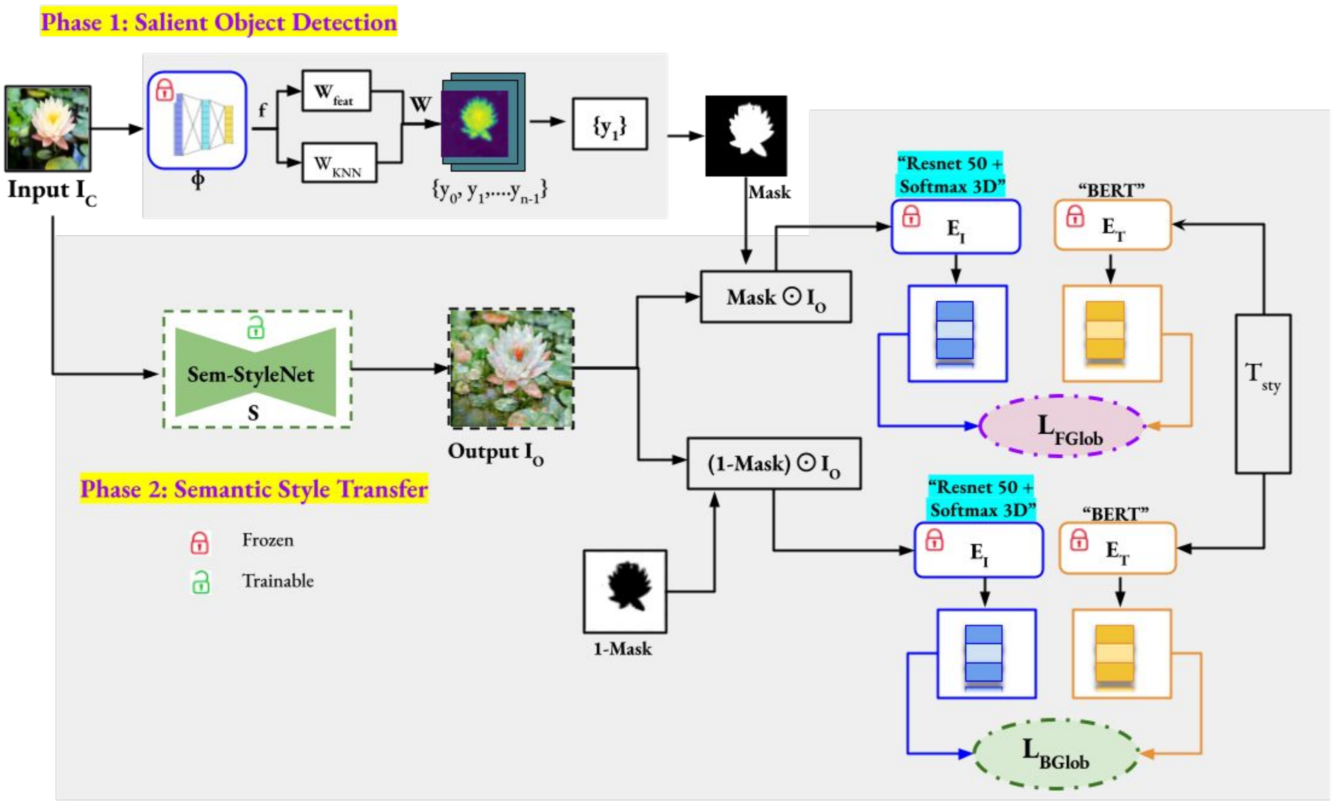

Phase1: Salient Object Detection
Input I_C
Deep Spectral Method [3]
Mask
T_{sty}
Phase2: Semantic Style Transfer

Method



Phase1: Salient Object Detection
Deep Spectral Method [3]
Input I_C
Mask
T_{sty}
Sem-StyleNet
S
Texture transfer on salient objects
Phase2: Semantic Style Transfer

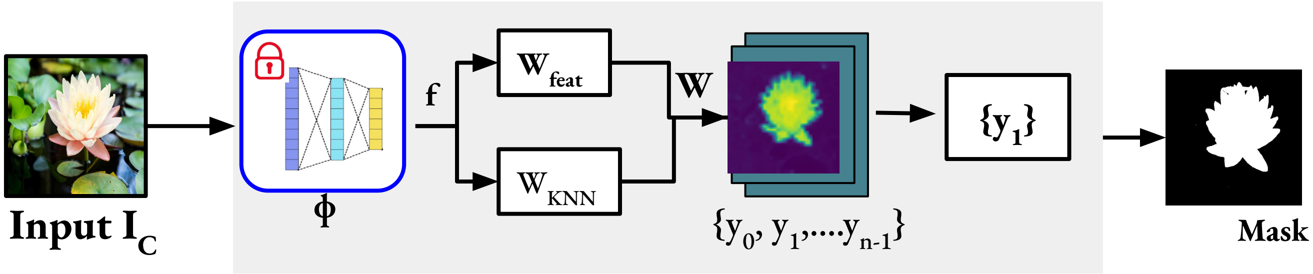
Method: Sem-CS



Notations

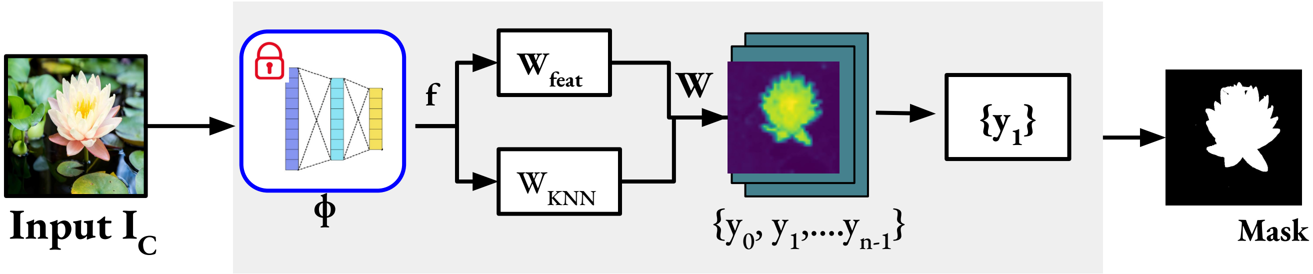
- I_C: Content Image
- T_{sty}: Style Text
- I_O: Stylized Output
- f: Deep patch features
- ϕ: Vision Transformer
- W_{KNN}: Color Matrix
- W_{feat}: Feature Matrix
- W: Semantic Affinity Matrix
- $\{y_0, y_1, \ldots, y_{n-1}\}$: Eigen vectors
- S: Semantic StyleNet
- $\odot$: Hadamard Product
- E_I: CLIP Image Encoder
- E_T: CLIP Text Encoder
- L_{BGlob}: Global background loss
- L_{FGlob}: Global foreground loss

Method



- I_C: Content Image
- ϕ: Vision Transformer
- f: Pre-trained dense features
- W_{feat}: feature Matrix
- W_{KNN}: Colour Affinity Matrix
- W: Semantic Affinity Matrix
- $\{y_0, y_1, \ldots y_{n-1}\}$: Eigenvectors

Method

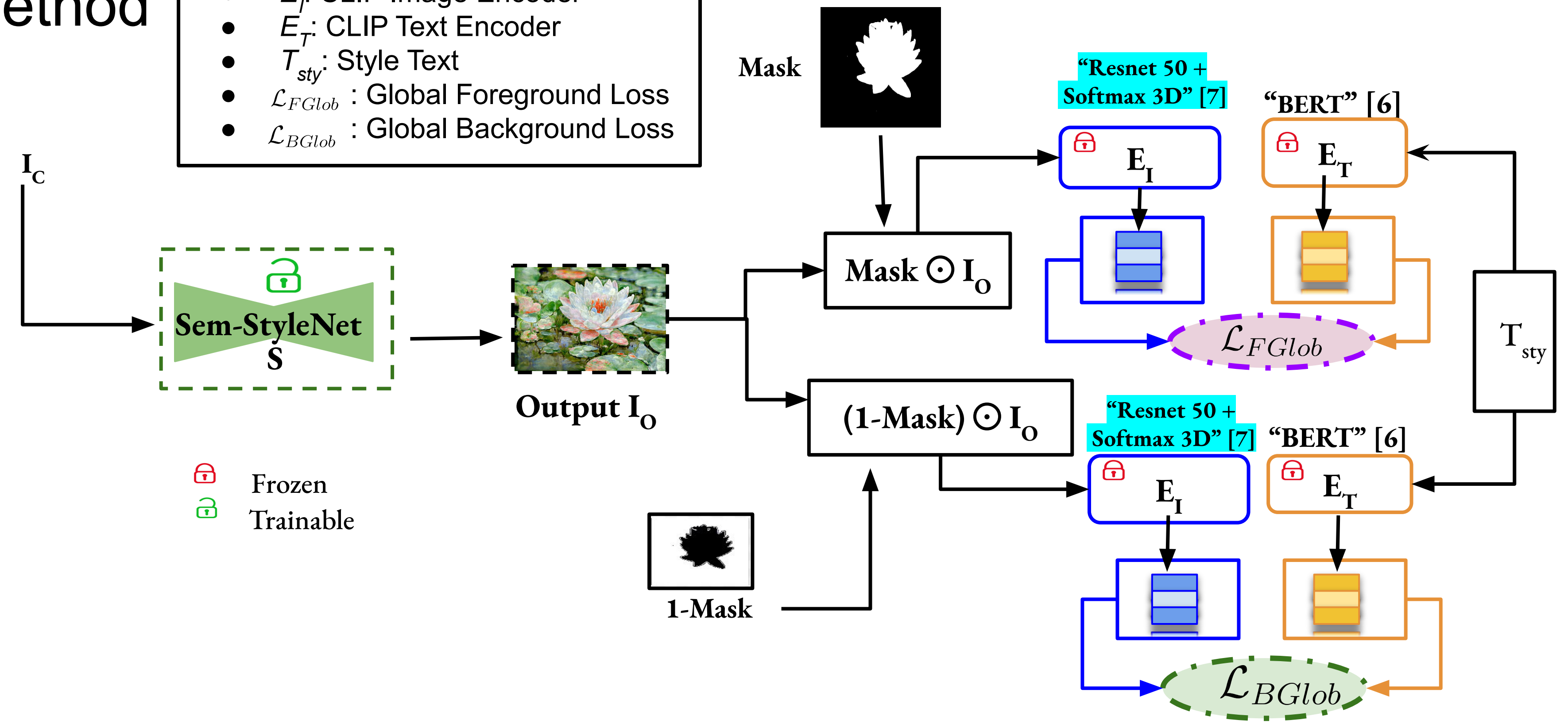


Phase1: Salient Object Detection

- I_C: Content Image
- ϕ: Vision Transformer
- f: Pre-trained dense features
- W_{feat}: feature Matrix
- W_{KNN}: Colour Affinity Matrix
- W: Semantic Affinity Matrix
- $\{y_0, y_1, \ldots y_{n-1}\}$: Eigenvectors

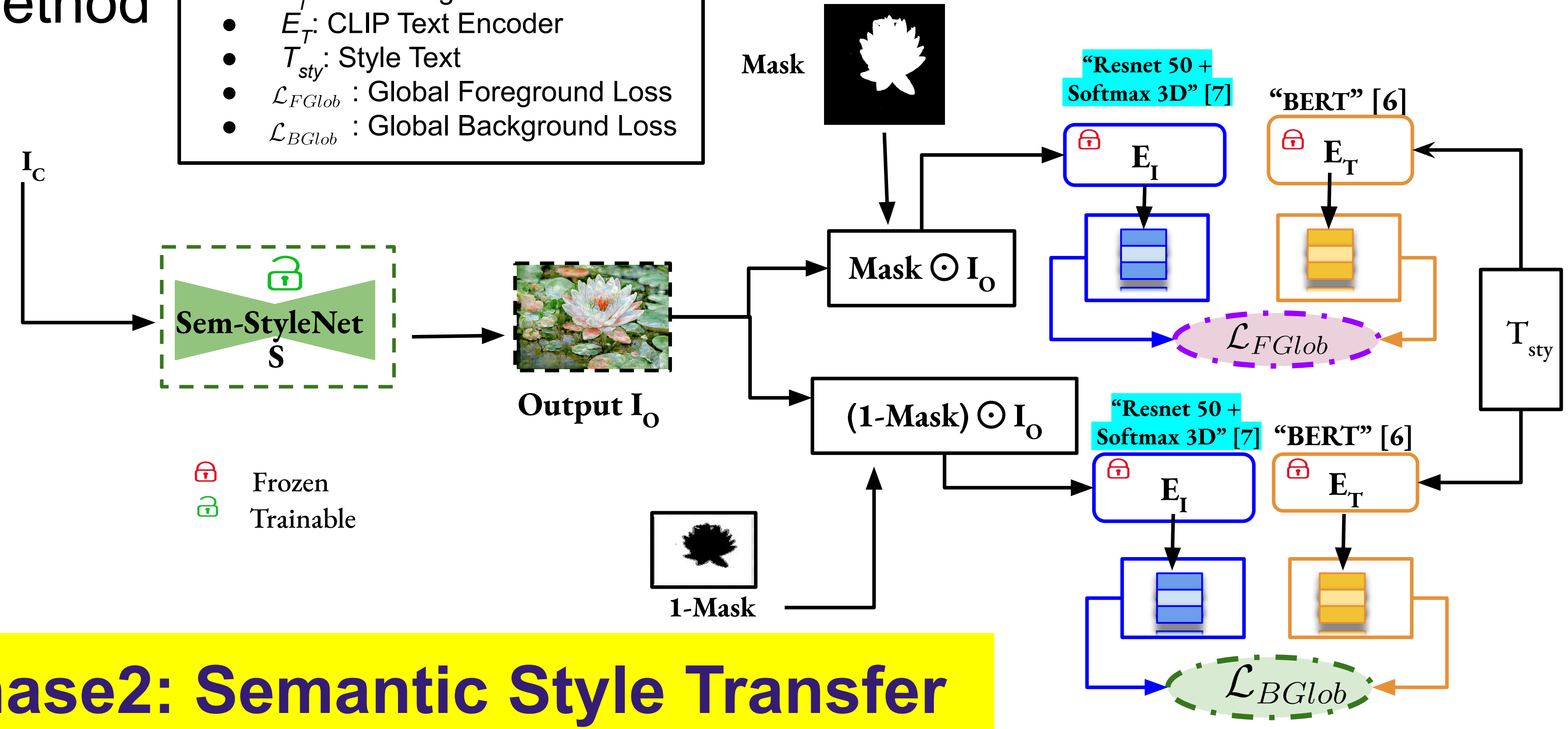
Method

- S: Semantic StyleNet
- I_o: Stylized Output
- E_I: CLIP Image Encoder
- E_T: CLIP Text Encoder
- T_{sty}: Style Text
- $\mathcal{L}_{FGlob}$: Global Foreground Loss
- $\mathcal{L}_{BGlob}$: Global Background Loss



Method

- S: Semantic StyleNet
- I_o: Stylized Output
- E_I: CLIP Image Encoder
- E_T: CLIP Text Encoder
- T_{sty}: Style Text
- $\mathcal{L}_{FGlob}$: Global Foreground Loss
- $\mathcal{L}_{BGlob}$: Global Background Loss



Phase2: Semantic Style Transfer

Algorithm: Semantic CLIPStyler

1: $\textbf{SEM-CS}(I_C, \phi, T_{sty}, E_T, E_I, S)$

- I_C: Content Image
- ϕ: Vision Transformer
- T_{sty}: Style Text
- E_T: CLIP Text Encoder
- E_I: CLIP Image Encoder
- S: Semantic StyleNet

Algorithm: Semantic CLIPStyler

1: $\textsc{Sem-CS}(I_C, \phi, T_{sty}, E_T, E_I, S)$

▷ *Compute Mask for salient objects identification*

2: $W = \text{AffinityMatrix}(I_c, \phi,)$

3: $\{y_0, y_1, \ldots, y_{n-1}\} = \text{Eigen\_Decomposition}(W)$

4: $Mask = \text{Extract\_Salient\_Object}(y_1)$

- I_C: Content Image
- ϕ: Vision Transformer
- T_{sty}: Style Text
- E_T: CLIP Text Encoder
- E_I: CLIP Image Encoder
- S: Semantic StyleNet

- $\{y_0, y_1, ..y_{n-1}\}$: Eigenvectors
- W: Semantic Affinity Matrix

Algorithm: Semantic CLIPStyler

$\textbf{Sem-CS}(I_C, \phi, T_{sty}, E_T, E_I, S)$

▷ *Compute Mask for salient objects identification*

$W = \text{AffinityMatrix}(I_c, \phi,)$

$\{y_0, y_1, \ldots, y_{n-1}\} = \text{Eigen\_Decomposition}(W)$

$Mask = \text{Extract\_Salient\_Object}(y_1)$

▷ *Perform Semantic Style Transfer*

$t_{fg}, t_{bg} = \text{Parse\_Style\_Text}(T_{sty})$

$I_{fg}, I_{bg} = Mask \odot S(I_C), (1 - Mask) \odot S(I_C)$

- I_C: Content Image
- ϕ: Vision Transformer
- T_{sty}: Style Text
- E_T: CLIP Text Encoder
- E_I: CLIP Image Encoder
- S: Semantic StyleNet

- $\{y_0, y_1, .. y_{n-1}\}$: Eigenvectors
- W: Semantic Affinity Matrix

- t_{fg}: Foreground Text embeddings
- t_{bg}: Background Text embeddings
- I_{fg}: Foreground Image embeddings
- I_{bg}: Background Image embeddings
- $\odot$: Hadamard Product

Algorithm: Semantic CLIPStyler

$\textsc{Sem-CS}(I_C, \phi, T_{sty}, E_T, E_I, S)$

▷ *Compute Mask for salient objects identification*

$W = \text{AffinityMatrix}(I_c, \phi,)$

$\{y_0, y_1, \ldots, y_{n-1}\} = \text{Eigen\_Decomposition}(W)$

$Mask = \text{Extract\_Salient\_Object}(y_1)$

▷ *Perform Semantic Style Transfer*

$t_{fg}, t_{bg} = \text{Parse\_Style\_Text}(T_{sty})$

$I_{fg}, I_{bg} = Mask \odot S(I_C), (1 - Mask) \odot S(I_C)$

▷ *Global Foreground Loss*

Compute Foreground Image Direction Loss $\Delta fg_I = E_I(I_{fg}) - E_I(I_C)$

Compute Foreground Text Direction Loss $\Delta fg_T \quad E_T(t_{fg}) - E_T(t_{src})$

$\mathcal{L}_{FGlob} = \text{Cosine\_similarity}(\Delta fg_I, \Delta fg_T) = 1 - \frac{\Delta fg_I . \Delta fg_T}{|\Delta fg_I||\Delta fg_T|}$

- I_C: Content Image
- ϕ: Vision Transformer
- T_{sty}: Style Text
- E_T: CLIP Text Encoder
- E_I: CLIP Image Encoder
- S: Semantic StyleNet

- $\{y_0, y_1,..y_{n-1}\}$: Eigenvectors
- W: Semantic Affinity Matrix

- t_{fg}: Foreground Text embeddings
- t_{bg}: Background Text embeddings
- I_{fg}: Foreground Image embeddings
- I_{bg}: Background Image embeddings
- $\odot$: Hadamard Product

- $\mathcal{L}_{FGlob}$: Global Foreground Loss

Algorithm: Semantic CLIPStyler

1: **SEM-CS**($I_C, \phi, T_{sty}, E_T, E_I, S$)

▷ *Compute Mask for salient objects identification*

2: W = AffinityMatrix($I_c, \phi,$)

3: $\{y_0, y_1, \ldots, y_{n-1}\}$ = Eigen_Decomposition(W)

4: $Mask$ = Extract_Salient_Object(y_1)

▷ *Perform Semantic Style Transfer*

5: t_{fg}, t_{bg} = Parse_Style_Text(T_{sty})

6: $I_{fg}, I_{bg} = Mask \odot S(I_C), (1 - Mask) \odot S(I_C)$

▷ *Global Foreground Loss*

7: Compute Foreground Image Direction Loss $\Delta fg_I = E_I(I_{fg}) - E_I(I_C)$

8: Compute Foreground Text Direction Loss $\Delta fg_T = E_T(t_{fg}) - E_T(t_{src})$

9: $\mathcal{L}_{FGlob}$ = Cosine_similarity($\Delta fg_I, \Delta fg_T$) $= 1 - \frac{\Delta fg_I . \Delta fg_T}{|\Delta fg_I||\Delta fg_T|}$

▷ *Global Background Loss*

10: Compute Background Image Direction Loss $\Delta bg_I = E_I(I_{bg}) - E_I(I_C)$

11: Compute Background Text Direction Loss $\Delta bg_T = E_T(t_{bg}) - E_T(t_{src})$

12: $\mathcal{L}_{BGlob}$ = Cosine_similarity($\Delta bg_I, \Delta bg_T$) $= 1 - \frac{\Delta bg_I . \Delta bg_T}{|\Delta bg_I||\Delta bg_T|}$

- I_C: Content Image
- ϕ: Vision Transformer
- T_{sty}: Style Text
- E_T: CLIP Text Encoder
- E_I: CLIP Image Encoder
- S: Semantic StyleNet

- $\{y_0, y_1, ..y_{n-1}\}$: Eigenvectors
- W: Semantic Affinity Matrix

- t_{fg}: Foreground Text embeddings
- t_{bg}: Background Text embeddings
- I_{fg}: Foreground Image embeddings
- I_{bg}: Background Image embeddings
- $\odot$: Hadamard Product

- $\mathcal{L}_{FGlob}$: Global Foreground Loss

- $\mathcal{L}_{BGlob}$: Global Background Loss

Algorithm: Semantic CLIPStyler

SEM-CS$(I_C, \phi, T_{sty}, E_T, E_I, S)$

▷ *Compute Mask for salient objects identification*

W = AffinityMatrix$(I_c, \phi,)$

$\{y_0, y_1, \ldots, y_{n-1}\}$ = Eigen_Decomposition(W)

$Mask$ = Extract_Salient_Object(y_1)

▷ *Perform Semantic Style Transfer*

t_{fg}, t_{bg} = Parse_Style_Text(T_{sty})

$I_{fg}, I_{bg} = Mask \odot S(I_C), (1 - Mask) \odot S(I_C)$

▷ *Global Foreground Loss*

Compute Foreground Image Direction Loss $\Delta fg_I = E_I(I_{fg}) - E_I(I_C)$

Compute Foreground Text Direction Loss $\Delta fg_T = E_T(t_{fg}) - E_T(t_{src})$

$\mathcal{L}_{FGlob}$ = Cosine_similarity$(\Delta fg_I, \Delta fg_T) = 1 - \frac{\Delta fg_I . \Delta fg_T}{|\Delta fg_I||\Delta fg_T|}$

▷ *Global Background Loss*

Compute Background Image Direction Loss $\Delta bg_I = E_I(I_{bg}) - E_I(I_C)$

Compute Background Text Direction Loss $\Delta bg_T = E_T(t_{bg}) - E_T(t_{src})$

$\mathcal{L}_{BGlob}$ = Cosine_similarity$(\Delta bg_I, \Delta bg_T) = 1 - \frac{\Delta bg_I . \Delta bg_T}{|\Delta bg_I||\Delta bg_T|}$

▷ *Minimize loss and compute output I_O*

$$I_O = \min_{\theta_S} \left(\mathcal{L}_{FGlob} + \lambda_{bg}\mathcal{L}_{BGlob}\right)$$

- I_C: Content Image
- ϕ: Vision Transformer
- T_{sty}: Style Text
- E_T: CLIP Text Encoder
- E_I: CLIP Image Encoder
- S: Semantic StyleNet

- $\{y_0, y_1, ..y_{n-1}\}$: Eigenvectors
- W: Semantic Affinity Matrix

- t_{fg}: Foreground Text embeddings
- t_{bg}: Background Text embeddings
- I_{fg}: Foreground Image embeddings
- I_{bg}: Background Image embeddings
- $\odot$: Hadamard Product

- $\mathcal{L}_{FGlob}$: Global Foreground Loss

- $\mathcal{L}_{BGlob}$: Global Background Loss

Algorithm: Semantic CLIPStyler

1: **SEM-CS**$(I_C, \phi, T_{sty}, E_T, E_I, S)$

▷ *Compute Mask for salient objects identification*

2: W = AffinityMatrix$(I_c, \phi,)$

3: $\{y_0, y_1, \ldots, y_{n-1}\}$ = Eigen_Decomposition(W)

4: $Mask$ = Extract_Salient_Object(y_1)

▷ *Perform Semantic Style Transfer*

5: t_{fg}, t_{bg} = Parse_Style_Text(T_{sty})

6: $I_{fg}, I_{bg} = Mask \odot S(I_C), (1 - Mask) \odot S(I_C)$

▷ *Global Foreground Loss*

7: Compute Foreground Image Direction Loss $\Delta fg_I = E_I(I_{fg}) - E_I(I_C)$

8: Compute Foreground Text Direction Loss $\Delta fg_T = E_T(t_{fg}) - E_T(t_{src})$

9: $\mathcal{L}_{FGlob}$ = Cosine_similarity$(\Delta fg_I, \Delta fg_T) = 1 - \frac{\Delta fg_I . \Delta fg_T}{|\Delta fg_I||\Delta fg_T|}$

▷ *Global Background Loss*

10: Compute Background Image Direction Loss $\Delta bg_I = E_I(I_{bg}) - E_I(I_C)$

11: Compute Background Text Direction Loss $\Delta bg_T = E_T(t_{bg}) - E_T(t_{src})$

12: $\mathcal{L}_{BGlob}$ = Cosine_similarity$(\Delta bg_I, \Delta bg_T) = 1 - \frac{\Delta bg_I . \Delta bg_T}{|\Delta bg_I||\Delta bg_T|}$

▷ *Minimize loss and compute output I_O*

13: $I_O = \min_{\theta_S} \left(\mathcal{L}_{FGlob} + \lambda_{bg}\mathcal{L}_{BGlob}\right)$

- I_C: Content Image
- ϕ: Vision Transformer
- T_{sty}: Style Text
- E_T: CLIP Text Encoder
- E_I: CLIP Image Encoder
- S: Semantic StyleNet

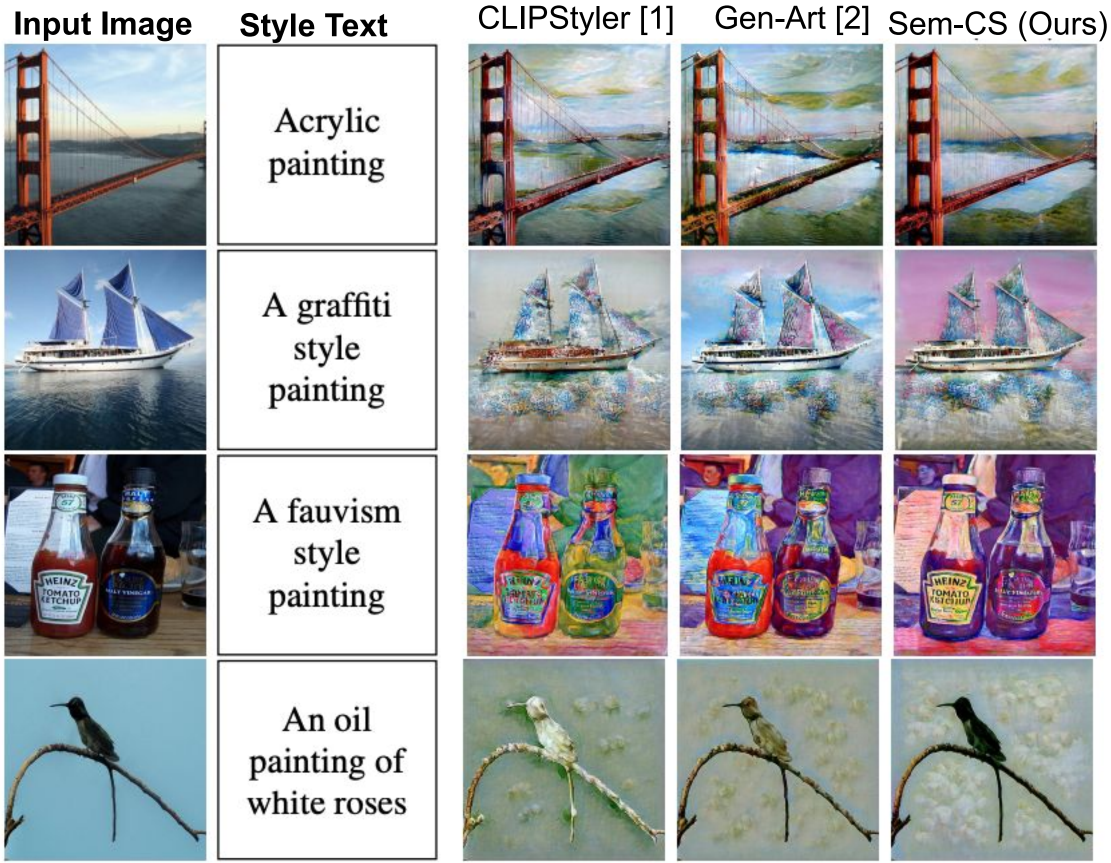
- $\{y_0, y_1, .. y_{n-1}\}$: Eigenvectors
- W: Semantic Affinity Matrix

- t_{fg}: Foreground Text embeddings
- t_{bg}: Background Text embeddings
- I_{fg}: Foreground Image embeddings
- I_{bg}: Background Image embeddings
- $\odot$: Hadamard Product

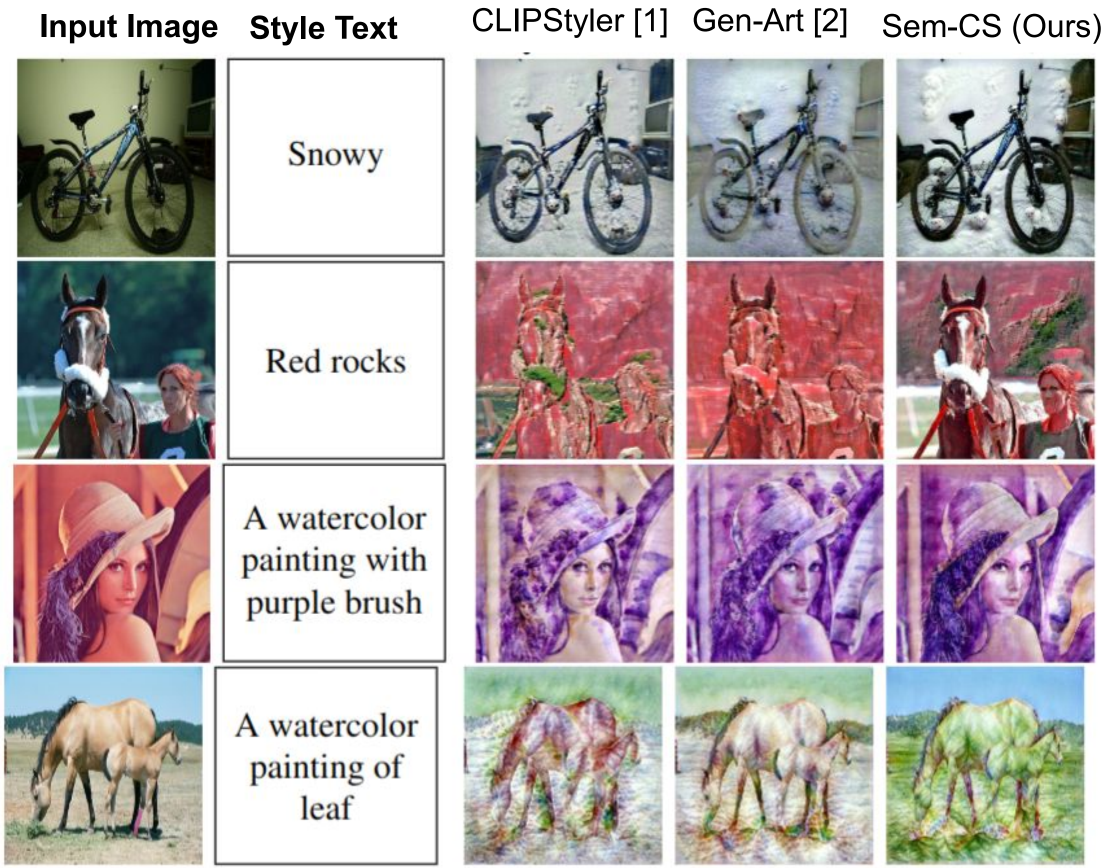
- $\mathcal{L}_{FGlob}$: Global Foreground Loss

- $\mathcal{L}_{BGlob}$: Global Background Loss

Results: Single Text Condition



Results: Single Text Condition



Results: Single Text Condition

Scores	CLIPStyler [1]	Gen-Art [2]	Sem-CS (Ours)
DISTS↑ [8]	0.32	0.25	**0.34**
NIMA↑ [9]	4.61	4.34	**5.34**
USer Study↑	28.3	33.1	**38.4**

Results: Double Text Condition

Style Text	Input Image	Gen-Art [2]	Sem-CS (ours)
F: Red Rocks **B**: Snowy			
F: Pop Art **B**: Starry Night By Vincent Van Gogh			

Results: Double Text Condition

Style Text

F: Red Rocks

B: Snowy

F: Pop Art

B: Starry Night By Vincent Van Gogh

Input Image **Gen-Art [2]** **Sem-CS (ours)**

Scores	Gen-Art [2]	**Sem-CS (Ours)**
DISTS↑ [8]	0.20	**0.33**
NIMA↑ [9]	4.24	**5.52**
User Study ↑	48.2	**51.7**

References [I]

1. Kwon, Gihyun, and Jong Chul Ye. "Clipstyler: Image style transfer with a single text condition." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
2. Yang, Zhenling, Huacheng Song, and Qiunan Wu. "Generative Artisan: A Semantic-Aware and Controllable CLIPstyler." arXiv preprint arXiv:2207.11598 (2022).
3. Melas-Kyriazi, Luke, et al. "Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

References [II]

4. Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
5. Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.
6. Kenton, Jacob Devlin Ming-Wei Chang, and Lee Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." Proceedings of naacL-HLT. Vol. 1. 2019.
7. Wang, Pei, Yijun Li, and Nuno Vasconcelos. "Rethinking and improving the robustness of image style transfer." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

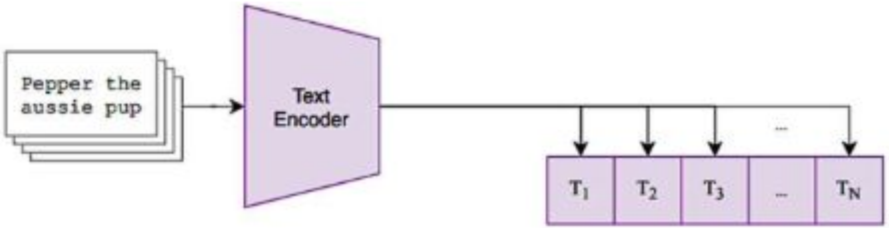
References [III]

8. Ding, Keyan, et al. "Image quality assessment: Unifying structure and texture similarity." IEEE transactions on pattern analysis and machine intelligence 44.5 (2020): 2567-2581.
9. Talebi, Hossein, and Peyman Milanfar. "NIMA: Neural image assessment." IEEE transactions on image processing 27.8 (2018): 3998-4011.

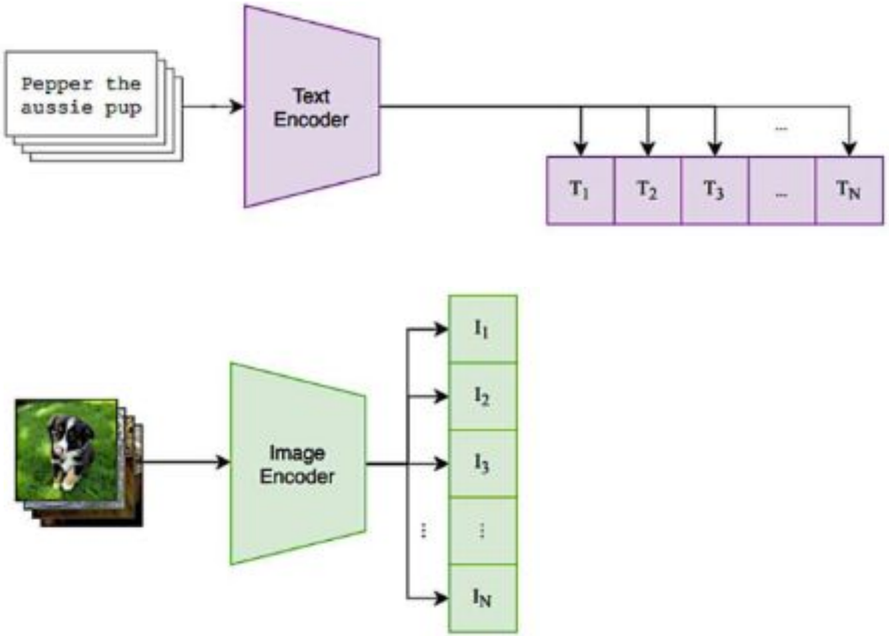
Thank You

Queries?

Background

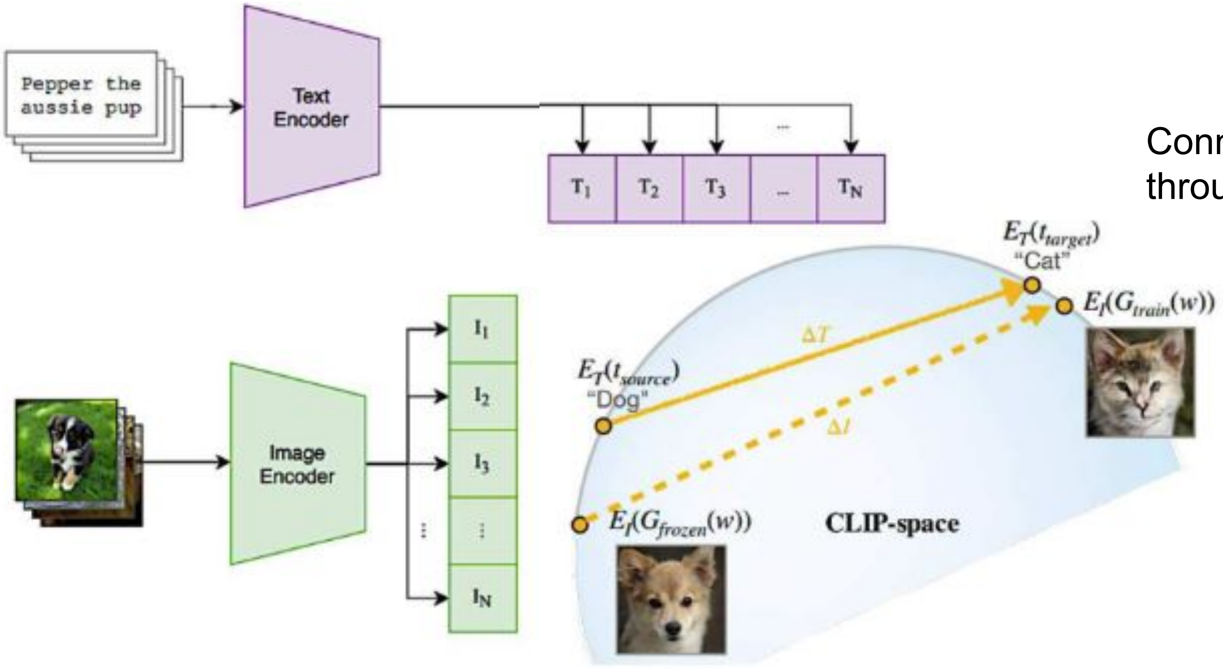


Background

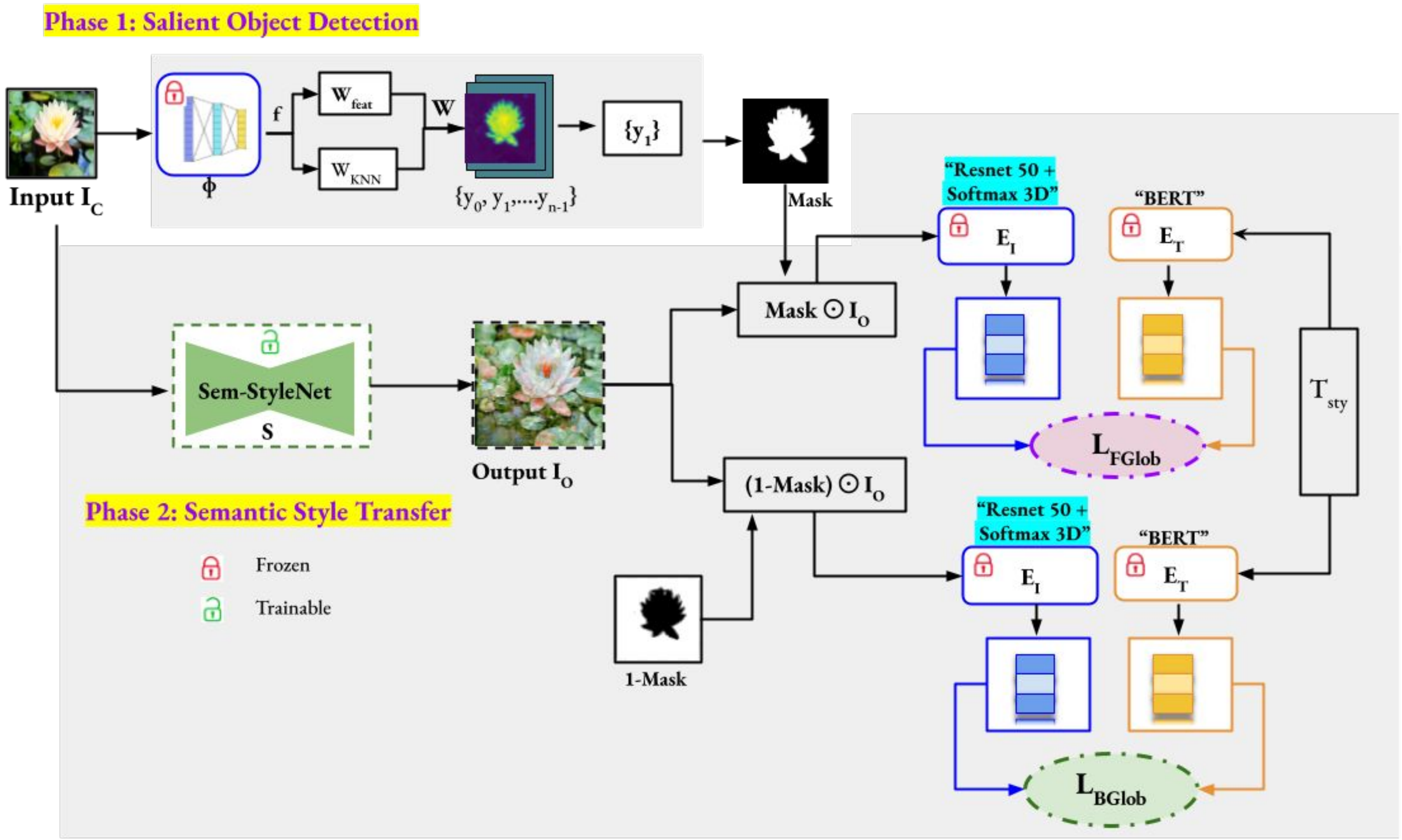


Connecting Text and Image through CLIP Space [5].

Background



Connecting Text and Image through CLIP Space [5].



Notations

- I_C: Content Image
- T_{sty}: Style Text
- I_O: Stylized Output
- f: Deep patch features
- ϕ: Vision Transformer
- W_{KNN}: Color Matrix
- W_{feat}: Feature Matrix
- W: Semantic Affinity Matrix
- $\{y_0, y_1, ..., y_{n-1}\}$: Eigen vectors
- S: Semantic StyleNet
- $\odot$: Hadamard Product
- E_I: CLIP Image Encoder
- E_T: CLIP Text Encoder
- L_{BGlob}: Global background loss
- L_{FGlob}: Global foreground loss