Walk into Virtual and Real World

Kuan-Chia Chen[1] Hyuntaek Oh[1] Woonki Kim[1]
[1] Department of Computer Science,Oregon State University
{chenku3,ohhyun,kimwoon}@oregonstate.edu

Abstract

Achieving real-time, object-aware video stylization remains an open challenge in computer vision. Most existing methods either fail to operate in real time or apply style effects indiscriminately, degrading the integrity of key foreground objects. In this paper, we propose a unified framework that combines Transformer-based Zero-Shot Video Object Segmentation (Isomer) with Fast Neural Style Transfer (FNST), allowing dynamic foreground preservation and background-specific stylization. Our method not only maintains temporal consistency and visual coherence across video frames but also achieves a practical processing speed of 15 FPS. This work bridges the gap between segmentation and stylization, offering an efficient solution for real-time creative video applications.

1. Introduction

Real-time video processing has recently become a crucial part of modern computer vision applications, including autonomous driving, augmented reality, and creative media production. Particularly, a challenging task in this domain is to simultaneously segment objects of interest and apply artistic transformations in real time. Traditional approaches often treat object segmentation and style transfer as independent problems, incurring inefficiency when attempting to combine them for video editing. Moreover, many existing methods either lack real-time performance or apply visual effects across the entire video frames, distorting foreground objects.

For context-aware and visually coherent video stylization, it is essential to separate foreground objects from the background and apply style transformations selectively. In applications such as augmented reality and film production, maintaining the integrity of key objects while modifying only the background improves both visual realism and creative control. However, achieving this goal in real time has several technical challenges: segmentation must operate without prior annotations, style transfer methods must maintain temporal consistency across frames to avoid flickering artifacts, and the combined system must remain computationally efficient.

Several methods have been proposed to address parts of this problem, such as using precomputed segmentation masks, applying optical flow or deploying Transformer-based models for object detection [4]. Similarly, Fast Neural Style Transfer (FNST)[8] can operate real-time stylization by replacing optimization with feed-forward networks. Yet, most FNST methods ignore object boundaries and apply style effects evenly, resulting in visually distracting results. Despite progress in segmentation and stylization separately, an efficient integration of both capabilities has still been underexplored for real-time video.

To bridge this gap, we propose a unified framework that integrates Zero-Shot Video Object Segmentation (ZVOS) with FNST to achieve real-time, object-aware video stylization. Our approach leverages Isomer[15], a Transformer-based segmentation model, to dynamically identify and preserve foreground objects without predefined annotation. Artistic style is then applied exclusively to the background using FNST, guaranteeing visual consistency and temporal coherence. Through extensive experiments, we demonstrate that our method not only preserves key object boundaries better than conventional methods but also achieves real-time performance, making it practical for interactive and creative video applications.

Altogether, the contributions of our work are as follows:

- We integrate Zero-Shot Video Object Segmentation with Fast Neural Style Transfer, guaranteeing real-time, background-specific stylization while preserving salient objects
- We provide high-speed processing and suitable quality, making it applicable to creative media and real-time video processing with object-aware transformations

2. Background

2.1. Zero-Shot Video Object Segmentation

Video Object Segmentation (VOS) is a fundamental task in computer vision that entails identifying objects and segmenting their boundaries across video frames. Traditional

methods depend on manual annotations, object tracking, or semi-supervised learning, typically requiring reference masks in the first frame [13]. This dependence on prior labels limits their practicality in real-time or open-world scenarios where object categories are unknown in advance.

To overcome these limitations, Zero-Shot Video Object Segmentation (ZVOS) has been proposed to segment salient objects without any form of supervision. ZVOS models infer object regions based on visual appearance and motion cues, making them more suitable for diverse and unseen objects. Recent Transformer-based architectures, such as Isomer [15], have shown promising results by effectively modeling semantic dependencies across frames, improving segmentation accuracy and computational efficiency in complex video scenes.

2.2. Fast Neural Style Transfer

Neural Style Transfer (NST) is a deep learning technique that transfers the artistic style of a reference image onto the content of another. Introduced by Gatys et al. [5], NST formulates this task as an optimization problem using a pretrained network such as VGG-19. It computes content loss to preserve spatial structure and style loss via Gram matrices to capture texture and patterns.

Fast Neural Style Transfer (FNST) improves efficiency by training a feed-forward network to apply stylization in a single forward pass [6], achieving real-time inference. Instead of optimizing each image individually, FNST learns a mapping that applies a specific style in a single forward pass. This approach is up to three orders of magnitude faster than traditional NST and is suitable for real-time applications. Some recent works have integrated Transformer-based architectures to improve spatial adaptability and content awareness [14]. However, most FNST models still apply style transformations uniformly across the entire frame, making them unsuitable for tasks that require selective stylization.

2.3. Transformer

Recent advances in Transformer architectures have significantly improved computer vision tasks by introducing self-attention mechanisms capable of modeling long-range dependencies across pixels [1]. Originally developed for natural language processing (NLP), Transformers have outperformed traditional Convolutional Neural Networks (CNNs) in tasks such as image segmentation, object detection, and feature extraction.

In Zero-Shot Video Object Segmentation (ZVOS), Transformers offer a key advantage over CNN-based models by capturing global context without requiring manual annotations. Their multi-head self-attention mechanism allows the model to relate distant regions within a frame, improving the ability to segment complex and previously unseen objects. The Isomer model [15], for instance, uses hierarchical Transformer blocks to refine object boundaries.

3. Related Work

3.1. Zero-Shot Video Object Segmentation

Early VOS approaches such as semi-supervised methods [10] depend on annotated reference frames, which limit their scalability in real-time applications. To reduce dependence, Convolutional Neural Networks (CNNs) methods [3] and recurrent architectures (RNNs) [12] were introduced for automatic object tracking. However, these methods often struggle with generalizing to unseen objects and dynamic environments.

Zero-Shot VOS (ZVOS) aims to segment objects without prior labels. Among recent approaches, Isomer [15] introduces a hierarchical Transformer-based framework that improves attention modeling and multi-scale feature fusion. Even though it is effective in segmentation tasks, these models are primarily evaluated on accuracy and have not been extended to applications such as selective visual stylization, missing an opportunity for methods that bridge segmentation with creative transformation tasks.

3.2. Fast Neural Style Transfer

Gatys et al. [5] first introduced NST as an optimization-based framework, producing high-quality results, but it suffers from high computational costs. To address this, Johnson et al. [8] proposed FNST, a feed-forward model that performs real-time style transfer. Despite its efficiency, it lacks spatial awareness and stylizes all regions uniformly, ignoring salient objects.

To mitigate this, AdaIN [7] introduced adaptive feature modulation for better content-style alignment, and Styleformer [14] applied attention mechanisms for improved adaptability. However, these approaches still lack region-specific control, limiting their utility in applications requiring selective or object-aware stylization.

3.3. Integration of Segmentation and Stylization

Several attempts have been made to combine segmentation with style transfer. For instance, Chen et al. [2] use saliency maps to preserve foreground regions, while Ma et al. [9] apply attention mechanisms for spatially localized style application. However, these methods often rely on manual masks or off-line processing, making them impractical for interactive video applications.

In contrast, we provide a novel unified framework that integrates ZVOS with FNST to achieve object-aware background stylization. By combining Isomer's hierarchical segmentation capabilities [15] with the efficiency of Transformer-based feed-forward style transfer [8], our

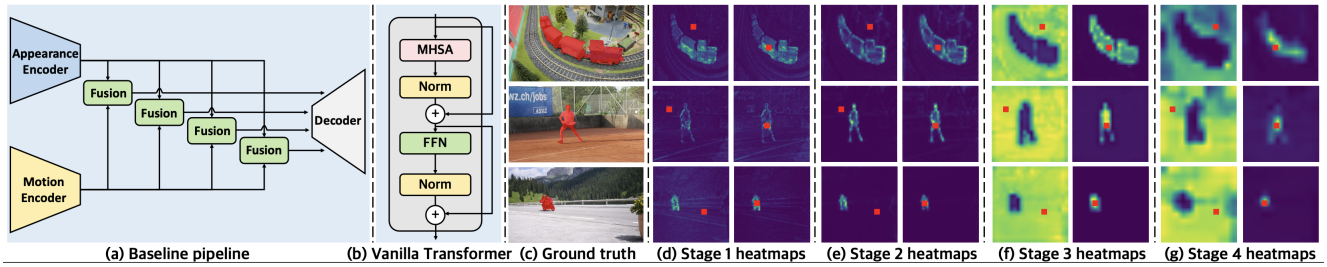


Figure 1. Vanilla Transformers attention map

method allows selective stylization while preserving foreground object integrity and maintains temporal consistency while operating at interactive speeds.

4. Method

4.1. Isomer

Isomer is an advanced Transformer-based framework specifically designed for efficient zero-shot video object segmentation (ZVOS). Inspired by attention observations from vanilla Transformers Fig.1. They purpose the structure Fig.2 which has two different functions: Context Sharing Transformer (CST) is for the first two stages, Semantic Gathering-Scattering Transformer (SGST) is for the last two stages.

4.1.1 Vanilla Transformer Baseline

The Vanilla Transformer Baseline integrates appearance and motion features effectively for tasks such as video segmentation. The proposed architecture incorporates specialized appearance and motion backbones, each designed to extract hierarchical features across four distinct stages from the current video frame and its corresponding optical flow map. At each stage, dedicated fusion modules systematically combine appearance and motion features channel-wise, effectively producing a unified mixed representation that captures rich, multi-model information. Subsequently, a sophisticated feature pyramid decoder integrates these fused representations from various stages, leveraging multi-scale information to enhance segmentation accuracy, ultimately generating the final segmentation prediction.

However, it has a substantial computational burden, significantly limiting practical applicability and inference speed.

4.1.2 Context Sharing Transformer

Context Sharing Transformer (CST) is good at computing global query-independent attention for all queries. It simplifies the multi-head self-attention step in a traditional Transformer by using global context modeling. The Global Context Modeling process is shared across all queries and operates first spatially, then channel-wise. At first, spatial attention generates a single-channel spatial attention map, which globally weights input features to increase context-relevant spatial regions. Then, channel attention refines this globally weighted context using convolutional layers interleaved with batch normalization and ReLU activation functions, effectively enhancing feature discrimination. The following step follows, incorporating a skip connection to merge the refined global context with the original input features, maintaining essential details. Finally, the integrated features are passed through the remaining Transformer layers, producing the final fused output representation.

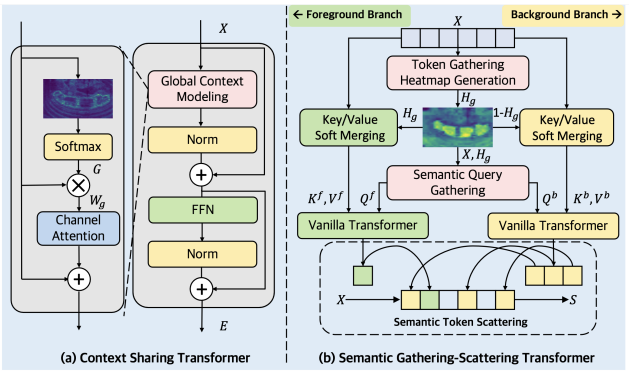


Figure 2. CST and SGST structure

4.1.3 Semantic Gathering-Scattering Transformer

Semantic Gathering-Scattering Transformer (SGST) can make models semantic dependencies between foreground and background features. It has two parallel branches, each processing foreground and background. The SGST first applies Semantic Query Gathering, which involves separating foreground tokens from the mixed input representations based on a predetermined threshold. Following this, the Key/Value Soft Merging step softly merges numerous tokens into a smaller set of representative tokens using a learned transformation matrix, reducing redundancy effectively. Subsequently, Dependencies Calculation is performed by applying a standard Transformer to efficiently model and capture semantic dependencies among these

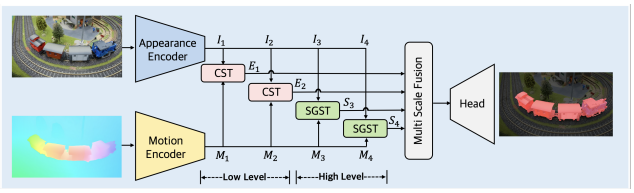


Figure 3. Isomer structure

compressed representative tokens. At the end, the Semantic Token Scattering step redistributes these updated semantic representations back to their original spatial positions, completing the semantic feature fusion process.

By the methods above, it dramatically reduces computations, making it have 87% less than the standard Transformer without sacrificing accuracy or performance.

4.1.4 Final Isomer Structure

The final structure of our proposed Isomer model is illustrated in Fig. 3. At first, the model takes appearance and motion data as input. These inputs are then processed by dedicated appearance and motion backbones, which extract hierarchical features across multiple stages. Secondly, at each stage, features from these two modalities are fused channel-wise, producing comprehensive mixed representations. Transformer-based fusion is then applied, utilizing CST modules for low-level stages and SGST modules for high-level stages. Afterward, the multi-stage fused representations are combined using a feature pyramid decoder, effectively integrating multi-scale contextual information. Finally, it will generate video object segmentation outputs.

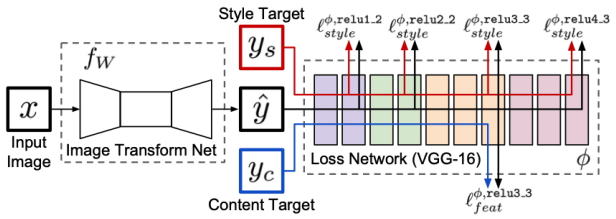


Figure 4. Style Transfer Structure

4.2. Style Transformer

In this work, we utilized a feed-forward approach for real-time style transfer by training a convolutional neural network with perceptual loss functions. Unlike optimization-based methods that iteratively refine an image to minimize a style loss proposed by Gatys et al., our approach enables direct transformation of an input image into a stylized output in a single forward pass[5].

Given an input image x, our goal is to generate an output image $\hat{y}$ that preserves the content structure of x while adopting the style characteristics of a reference style image y_s. To achieve this, we train a transformation network $f_W(x)$, parameterized by weights W, to learn this mapping.

Our framework consists of two main components:

1. Image Transformation Network $f_W(x)$: A deep convolutional neural network that takes an input image and produces a stylized output.

2. Loss Network ϕ: A pre-trained VGG-16 network, used to compute perceptual losses that measure content and style differences.

4.2.1 Image Transformation Network

The Image Transformation Network $f_W(x)$ is a feed-forward convolutional neural network that stylizes an input image in a single pass. It follows an encoder-transformer-decoder architecture that compresses the input, applies style transformations via residual blocks, and reconstructs the output at the original resolution, balancing speed and visual quality.

The encoder uses stride-2 convolutions to downsample the image, capturing broad contextual features while reducing computational cost. These early layers extract low-level features such as edges and textures.

At the network's core are five residual blocks, each containing two 3×3 convolutional layers with batch normalization and ReLU activations. Residual connections help preserve content and facilitate learning identity mappings, making them well-suited for style transfer tasks.

The decoder upsamples the transformed features back to the original resolution using fractionally-strided (stride-0.5) convolutions. A final activation ensures output pixel values fall within a valid range.

Ultimately, the network learns to produce a stylized output $\hat{y} = f_W(x)$ that maintains the content of the input while adopting the style of a reference image. This is guided by perceptual loss functions computed using a separate, pre-trained network.

4.2.2 Loss Network

Instead of relying on per-pixel losses, we leverage high-level feature representations extracted from the loss network ϕ. We define two perceptual loss terms:

1. Content Loss: Ensures the output image preserves the semantic content of the input image by minimizing the feature reconstruction error at a chosen layer j of ψ:

$$\mathcal{L}_{\text{content}}(\hat{y}, x) = \frac{1}{Z_l} \|\psi_l(\hat{y}) - \psi_l(x)\|_2^2$$

where $Z_l = C_l \cdot H_l \cdot W_l$ is the total number of features at layer l.

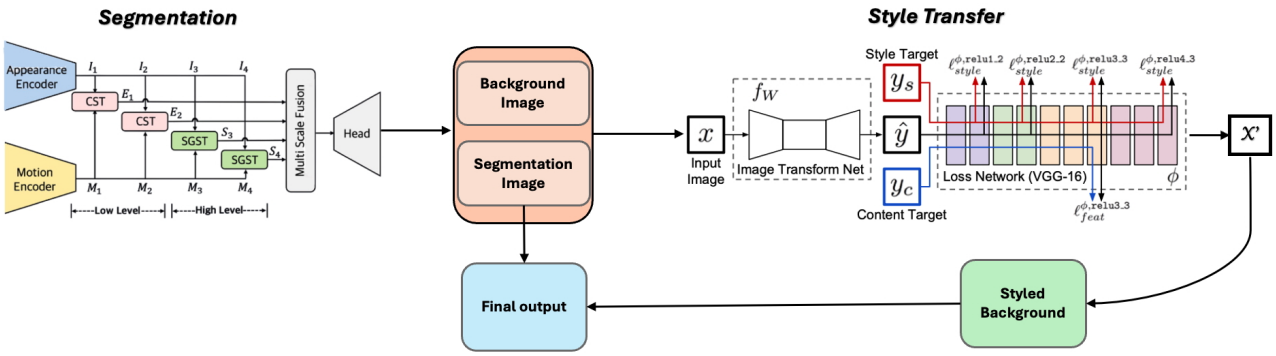


Figure 5. Final network Structure

2. Style Loss: Encourages the output image to match the style of y_s by minimizing the difference in Gram matrices, which capture feature correlations at multiple layers J in ψ:

$$\mathcal{L}_{\text{style}}(\hat{y}, x_s) = \sum_{l \in \mathcal{S}} \|\Gamma(\psi_l(\hat{y})) - \Gamma(\psi_l(x_s))\|_F^2$$

where $\Gamma(F) = \frac{1}{C_l H_l W_l} F F^\top$ and F is the reshaped feature map at layer l, flattened across spatial dimensions.[8].

4.2.3 Training Objective

The transformation network is optimized to minimize a weighted combination of content loss, style loss, and total variation regularization:

$$W^* = \arg\min_W \mathbb{E}_{x \sim \mathcal{X}} \left[\alpha \mathcal{L}_{\text{content}} + \beta \mathcal{L}_{\text{style}} + \delta \mathcal{L}_{\text{Reg}}\right]$$

where α, β, and δ are weighting factors that control the influence of the content, style, and regularization losses, respectively. By adjusting the values of α and β, we can guide the model to prioritize either content preservation or stylistic fidelity, depending on the desired outcome [8].

4.3. Final Model (Isomer + Styled model)

Combining the two separate methods, we propose a method that introduces an object-preserving style transfer framework, which maintains the integrity of the primary object while applying artistic transformations exclusively to the background. This is achieved through a three-step process: segmentation, style transfer, and reconstruction as shown in 5. Each stage is designed to ensure efficient processing while preserving content structure and visual consistency.

4.3.1 Segmentation

The first stage of our method involves separating the object from the background using advanced segmentation techniques. A deep learning-based segmentation model is employed to generate an accurate mask that distinguishes the foreground object from the rest of the image. This ensures that the style transfer operation is applied only to the background, preventing distortions in the object. The segmentation model is optimized to provide high precision while maintaining real-time processing capability.

However, in video or webcam input, the initial optical flow does not exist at the first frame, which is required for motion-based segmentation. To address this, we initialize the flow using the Farneback optical flow algorithm, which estimates dense motion between consecutive frames. This allows the segmentation model to receive meaningful motion cues even from the beginning, ensuring smooth and accurate object-background separation throughout the video sequence.

4.3.2 Stylizing and Reconstruction

Once the background has been successfully isolated through segmentation, the style transfer process applies artistic transformation to the entire frame rather than just the background. The original frame is first converted into a tensor representation before being passed through a pretrained Transformer Network, which stylizes the entire image in a single forward pass. This ensures efficient real-time processing, making the approach suitable for video applications. The transformation is guided by perceptual loss functions, which help maintain structural details while altering the colors and textures according to the selected artistic style. Different style models can be dynamically loaded to provide various artistic effects.

However, to preserve the integrity of the foreground ob-

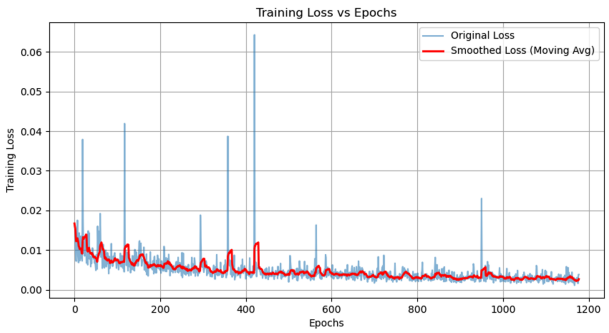
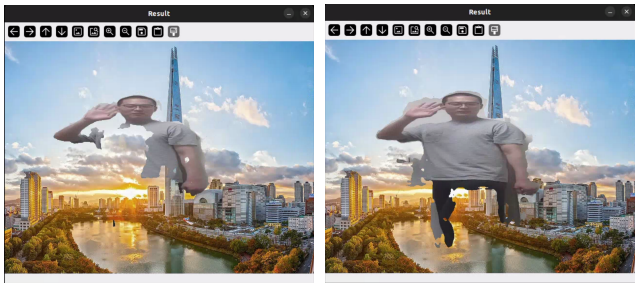


Figure 6. Isomer Segmentation loss



(a) Isomer pretrain model test in video

(b) Isomer finetune model test in video

Figure 7. Comparison of Isomer pretrain and finetune models

ject, the segmentation mask generated in the earlier step is used to reveal the original object over the stylized image. While the whole frame undergoes style transfer, the segmentation mask—derived from the object detection model—remains untouched. This mask is then applied over the stylized frame, restoring the original object while keeping the stylized background intact. By following this structured approach, we ensure that the artistic transformation enhances the scene without distorting the primary object. This technique enables seamless integration of artistic styles in real-time video streams, balancing creativity with realism.

5. Experiment

We initially trained the Isomerous Segmentation Network and the Style Transfer Network independently, each utilizing its respective dataset to ensure optimal performance in their respective tasks.

Once both models were pre-trained, we integrated them into a unified framework, leveraging their learned representations to create a seamless segmentation-driven style transfer pipeline. The combined model was then evaluated across image, video, and real-time webcam inputs to assess its operational effectiveness and ensure its robustness in diverse real-world scenarios.

5.1. Isomer

In the early stage, we would like to try some new technology. Then we got some good results from experiencing big models such as SAM 2: Segment Anything in Images and Videos[11] and HQTrack[16]. However, we need accuracy and small running time in our work. Additionally, those models require large VRAM for finetuning. Therefore, we found Isomer, which requires lower VRAM and has around 15 fps when we infer it.

For Isomer, we trained the model using a diverse set of datasets, including DAVIS (human dataset), FlyingChairs (FBMS), KITTI 2012 & 2015, MPI Sintel, Spring, and Multi-Human Data Generation. These datasets were specifically selected for their focus on human-centric segmentation with optical flow, aligning with our objective of separating human subjects from the background.

After training with these datasets, we fine-tuned the Isomer pretrained model on the DAVIS human dataset. While Isomer is originally designed for segmentation across multiple object classes, this fine-tuning step enhances its ability to distinguish human figures more accurately. By leveraging these datasets, the model learns precise segmentation of human subjects, ensuring the background can be stylized independently while preserving the integrity of human features.

For the first step, we try to finetune the Isomer pretrained model with the backbone ”swin tiny” model, then training with a large number of classes dataset. The performance of the first step doesn't work well. In our step 2, we scale down the training dataset and only give it the images with humans inside. The dataset is 21 classes, 937 different images, flow maps, and ground truth. We train on RTX 4090 GPU for 1200 epochs with the batch size 8. The training result is shown in Fig. 3.

5.2. Style Transfer

Our style transfer network was trained on the Microsoft COCO dataset, which contains 80,000 images with diverse objects and textures. Each image was resized to 256 × 256 , and the model was trained with a batch size of 4 for 40,000 iterations , covering approximately two epochs over the dataset.

We employed the Adam optimizer with a learning rate of 10^{-3} . To maintain smoothness in the generated outputs, we applied total variation regularization , with a strength between 10^{-6} and 10^{-4} , selected via cross-validation for each style target. For loss computation, we utilized feature reconstruction loss at layer relu2_2 and style reconstruction loss at layers relu1_2, relu2_2, relu3_3, and relu4_3 of the VGG-16 loss network. Our implementation was built using PyTorch and the training process took about 1 hour on a single NVIDIA RTX 4090 GPU.

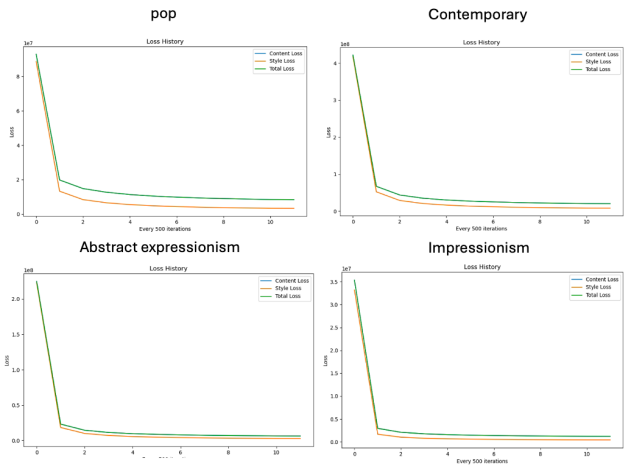


Figure 8. Style Transfer loss on variety style image

As illustrated in Figure 8, the style loss was computed for different styles, demonstrating a strong convergence trend. Notably, all variations converged effectively, with losses approaching zero , indicating stable and consistent optimization across different styles.

5.3. Final Model (Isomer + Styled model)

We first tested various artistic styles on a diverse set of images with flows. As shown in Figure 9, the results demonstrate sophisticated segmentation and precise styling, effectively distinguishing the foreground from the background. The segmentation model ensures that the object remains intact while the background has been stylized, without unwanted distortions. This validates the robustness of our approach in maintaining structural integrity while achieving high-quality stylization.

Then to test the practicality, we tested our method on a live webcam feed, achieving an average of 15 FPS using an NVIDIA RTX 4090 GPU. To further evaluate its performance across different scenarios, we replaced the original background with a separate video, simulating dynamic environments. As shown in Figure 10, the system effectively separates the human object from the background while applying style transfer seamlessly to the background. The segmentation mask ensures that the foreground remains intact, preserving the natural appearance of the subject.

The achieved 15 FPS demonstrates that our approach is practical for real-time applications, making it suitable for scenarios where background replacement and stylization are needed, such as virtual conferencing, live streaming, or augmented reality. The combination of efficient segmentation and fast style transfer enables smooth and visually consistent transformations without significant delays.

Moreover, by leveraging various pretrained style transfer models, we enabled dynamic style changes in real time while running a webcam. Additionally, our approach allows users to selectively apply style transfer in three different modes:

(1) stylizing only the background

(2) stylizing only the object

(3) stylizing the entire image

This flexibility highlights the adaptability of our method, making it well-suited for a wide range of applications, including virtual production, creative content generation, and interactive augmented reality experiences.

6. Conclusion

Our goal of developing a deep neural network capable of segmenting objects while dynamically applying style transfer to the background has been successfully achieved. By leveraging optical flow-based segmentation and pretrained style transfer models, our method enables real-time artistic transformations while preserving the integrity of the foreground object.

Through extensive testing on both static images and real-time webcam input, we demonstrated that our approach achieves an average of 15 FPS on an NVIDIA RTX 4090, making it practical for real-time applications. The ability to dynamically control the stylization—whether applied to the background, the object, or the entire image—further enhances the versatility of our system.

Our work opens new possibilities for applications in virtual conferences, augmented reality, creative media, and interactive video processing.

7. Future Work

While our current implementation demonstrates promising results in real-time performance and visual quality, there are several directions for future improvement and exploration.

First, the segmentation and style transfer processes currently operate as separate modules, which introduces additional latency. A key area for future work is to design an end-to-end architecture that fuses segmentation and stylization into a single unified model. This integration could lead to improved efficiency, lower memory usage, and potentially higher frame rates.

Another potential research direction involves further optimizing CST and SGST to enhance inference speed. While our current implementation has achieved notable performance improvements, it runs at approximately 15 frames per second (FPS) on a high-end GPU such as the RTX 4090. This hardware requirement might not be practical or accessible for all users. Therefore, continuing research efforts to develop more computationally efficient versions of CST and

(a) Parkour

(b) Pop Art

(c) Paragliding

(d) Korean Pattern

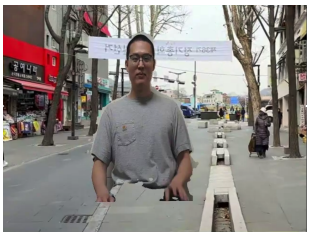
(e) Motorcycle

(f) Contemporary Art

(g) Kite Surf

(h) Dragonball Style

Figure 9. Comparison of Original and Stylized Images in a 4×3 Layout



(a) Original Video Frame

(b) Stylized Video Frame

Figure 10. Comparison of Original and Stylized Backgrounds.

SGST could broaden their applicability and usability across various hardware platforms.

References

[1] Recent Advances in Vision Transformer: A Survey and Outlook of Recent Work.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[3] S. Dutt Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[4] L. Fan, T. Zhang, and W. Du. Optical-flow-based framework to boost video object detection performance with object enhancement. *Expert Systems with Applications*, 170:114544, 2021.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[6] D. Holden, I. Habibie, I. Kusajima, and T. Komura. Fast neural style transfer for motion data. *IEEE Computer Graphics and Applications*, 37(4):42–49, 2017.

[7] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[8] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711, 2016.

[9] Z. Ma, J. Li, N. Wang, and X. Gao. Semantic-related image style transfer with dual-consistency loss. *Neurocomputing*, 406:135–149, Sept. 2020.

[10] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation. pages 724–732, 2016.

[11] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[12] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[13] W. Wang, J. Shen, F. Porikli, and R. Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):985–998, 2019.

[14] X. Wu, Z. Hu, L. Sheng, and D. Xu. Styleformer: Real-time arbitrary style transfer via parametric style composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14618–14627, October 2021.

[15] Y. Yuan, Y. Wang, L. Wang, X. Zhao, H. Lu, Y. Wang, W. Su, and L. Zhang. Isomer: Isomerous transformer for zero-shot video object segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 966–976, 2023.

[16] J. Zhu, Z. Chen, Z. Hao, S. Chang, L. Zhang, D. Wang, H. Lu, B. Luo, J.-Y. He, J.-P. Lan, et al. Tracking anything in high quality. *arXiv preprint arXiv:2307.13974*, 2023.