Approaches in Neual Style Transfer: VGG19 and Vision Transformers

(2024F) COMP-5112-FA - Research Methodology

Junjun Hu
1264029
Lakehead University
Thunder Bay, Canada
jhu29@lakeheadu.ca

Zhijie Shen
1263245
Lakehead University
Thunder Bay, Canada
zshen9@lakeheadu.ca

Dr. Jinan Fiaidhi
Faculty
Lakehead University
Thunder Bay, Canada
jfiaidhi@lakeheadu.ca

***Abstract*—Neural Style Transfer (NST) has emerged as a groundbreaking technique that combines the structural content of one image with the artistic style of another, producing visually compelling results which is widely used in image processing, content creation and digital art. Since its inception, NST has undergone significant advancements, evolving from computationally intensive algorithms to efficient real-time applications. This paper explores the comparative performance of two prominent models, VGG19 and Vision Transformers (ViT), in the context of NST. We evaluate the models across various metrics, including content fidelity, style adaptation, image quality and computational efficiency. Experimental results demonstrate that while ViT offers computational advantages and noise reduction, VGG19 outperforms in delivering artistically coherent and visually appealing outputs. This study shows the trade-offs between traditional convolutional approaches and transformer-based methods, providing valuable insights into their applicability for different use cases in style transfer.**

***Index Terms* Terms—Neural Style Transfer, Convolutional Neural Networks, Vision Transformers, VGG19, Artistic Style Transfer, Self-Attention Mechanisms, Computational Efficiency, Real-Time Applications.**

I. Introduction

NEURAL Style Transfer (NST) is a groundbreaking deep learning technique that combines the content of one image with the artistic style of another, enabling the creation of unique, visually compelling works of art [1]. First introduced by Gatys et al. in 2015, NST employs Convolutional Neural Networks (CNNs) to analyze and synthesize content and style features. This innovative approach has revolutionized image editing and creative design, transforming ordinary photos into extraordinary artistic representations [2].

Deep learning, inspired by the human brain's neural structure, automates feature extraction from large datasets. CNNs, a specialized form of deep learning model, are particularly effective in image processing due to their layered architecture, which captures hierarchical features. These features range from simple edges in the initial layers to complex patterns in deeper layers. The VGG network, a prominent CNN architecture, is widely utilized for NST due to its efficiency in multi-layered feature extraction and integration [4].

While CNNs have dominated NST implementations, the introduction of Vision Transformers (ViT) has marked a paradigm shift. Unlike CNNs, ViT utilizes self-attention mechanisms to analyze global relationships within an image, making it highly effective for capturing intricate patterns and textures [3]. By dividing images into patches and processing them through attention layers, ViT achieves superior contextual understanding, offering new possibilities for style and content synthesis.

This paper explores the advancements in NST, focusing on the comparative performance of VGG19 and ViT in various aspects such as style preservation, computational efficiency, and user satisfaction. The study highlights the strengths and limitations of these models, offering insights into their applications for artistic transformations. By addressing the trade-offs between traditional CNN-based approaches and transformer-based innovations, this research contributes to the broader understanding and optimization of NST techniques [5].

II. LITERARY REVIEW

Neural Style Transfer has evolved significantly since its inception, with key developments driving its progress. In 2015, Gatys et al. laid the foundation for NST with their work "A Neural Algorithm of Artistic Style." They used CNNs to extract content and style features from images and combined them through an optimization process to create a new image. This marked the first use of deep learning for artistic style transfer. Building on this, Johnson et al. in 2016 introduced a real-time style transfer method. By using perceptual loss functions, they accelerated the style transfer process, making it feasible for mobile applications and real-time deployment. Further advancements came in 2018 with Fast Neural Style Transfer,

which reduced computational costs while maintaining output quality. This innovation expanded the practical applications of NST, making it accessible for broader use cases. More recently, Togo et al. in 2021 introduced text-guided style transfer, allowing users to specify desired styles through text descriptions. This method enhanced the flexibility and accessibility of NST, enabling more personalized and creative outputs. These developments highlight NST's continuous progress, from improving computational efficiency to broadening its application scope. Meanwhile, the integration of Vision Transformers demonstrates the latest advancements, providing enhanced capabilities for style and content synthesis.

III. Methodology

A. Dataset

The data sets used in this experiment include pre-trained models and enhanced image sets. The VGG19 model is pre-trained on the ImageNet dataset, which contains more than 14 million images across 1,000 categories. The images were resized to 224*224 pixels to meet the input requirements
CIFAR-100 dataset consists of 60,000 color images,each of size 32*32 pixels,divided into 100 classes,each class contains 600images and futher grouped into 20 superclasses,with 5 subclasses per superclass.The training set is 50,000 images,and the test set consists of 10,000 images.

This dataset provides powerful initialization capabilities for extracting hierarchical features required for neural style transfer tasks. In addition to ImageNet, custom content and style images are curated to evaluate the model's performance in different scenarios.The content images are selected from high-resolution landscape photographs of Thunder Bay and represent structural elements. These images were chosen because they have clearly recognizable features that make it easy to save content during style transfer. The style images are inspired by the artwork of the Thunder Bay Art Gallery, which is known for showcasing Aboriginal art and regional culture. These images reflect complex textures and patterns and are ideal for evaluating the style capabilities of NST models.

B. Environment

The development environment utilized Envirnoment Setup.for the following requirement and framework.

For the **hardware** The GPU recommended NVIDIA GPU especially for the VGG and ViT models.The memory at least 8GB of RAM.

For the **software**:The implementation was conducted in a locally configured environment using Python version 3.9 with TensorFlow 2.9.1, Keras 2.9.0, OpenCV 4.6.0, Scikit-learn 1.0.2, NumPy,Pytorch 2.3.2 Pandas, Matplotlib, and SciPy, developed in Visual Studio with GPU acceleration enabled through NVIDIA CUDA Toolkit 12.3. and use Jupyter Notebook.

C. Overall Model Design

Vision Transformers (ViT):

Transformers have been a significant innovation in the field of computer vision in recent years, initially proposed by Vaswani et al. in 2017, primarily for natural language processing (NLP) tasks. As research progressed, Transformer was gradually applied to other areas, such as computer vision and image generation. In this study, we use the Vision Transformer (ViT), a Transformer model specifically designed for processing images.

Vision Transformers (ViT) represent a paradigm shift in image processing by using self-attention mechanisms instead of traditional convolutional operations. The ViT model first divides the input image into small patches and linearly embeds them, then processes these embeddings through transformer encoder layers. Using positional encoding, ViT can capture the relative positional relationships within the image, which helps in the blending of content and style. In our style transfer task, we used a pre-trained ViT model on ImageNet and fine-tuned it to adapt to the specific style transfer task. Specifically, we selected multiple transformer encoder blocks to compute content and style losses, achieving a hierarchical representation of the image. This method allows ViT to effectively capture style features while retaining the content structure.

For style feature extraction, we used multiple encoder layers to obtain different levels of style information, followed by a decoder module to reconstruct the image. This approach allows us to effectively combine multi-level features from the content and style images, ultimately generating an image with the desired style.

Figure 1 shows the architecture of Vision Transformer (ViT) for Neural Style Transfer (NST).

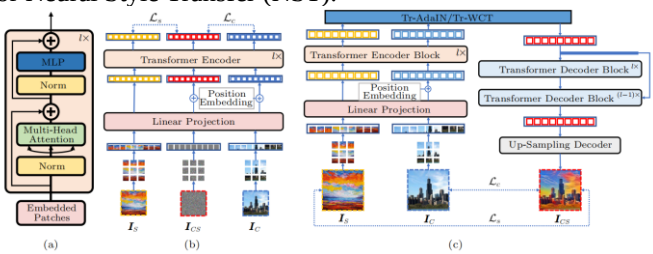

Fig. 1. The architecture of Vision Transformer (ViT) for Neural Style Transfer (NST). (a) The transformer encoder block shows the process of embedded patches being passed through multi-head attention and feed-forward layers. (b) The image patches from the content image (Ic) and style image (Is) are projected through a linear projection layer, followed by transformer encoder blocks. Content loss (Lc) and style loss (Ls) are calculated to optimize the style transfer. (c) The decoder part reconstructs the stylized image (Ics) using the combined features of content and style [3].

VGG19:

VGG19 is a Convolutional Neural Network (CNN) model that uses small 3x3 convolutional kernels. The model is well-suited for extracting different levels of image features, which is critical for content preservation and style blending in NST.

VGG19 consists of 19 layers, including convolutional layers, pooling layers, and fully connected layers. The architecture starts with five blocks, each containing multiple convolutional layers followed by a max-pooling layer. These convolutional layers are used to extract spatial features from the input images, while the pooling layers help reduce the spatial dimensions and retain important features. The fully connected layers at the end are responsible for classification tasks, in NST, we focus primarily on the feature extraction capability of the earlier convolutional layers.

Figure 2: The architecture of VGG19 illustrates the process by which an input image is passed through the series of

convolutional and pooling layers to extract essential features. This feature extraction plays a crucial role in identifying both content and style features from the respective images.

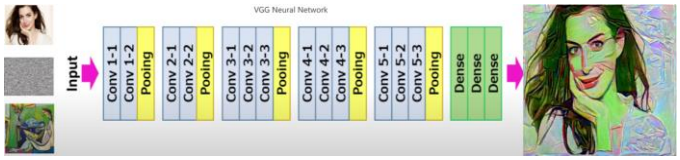

Fig. 2. The VGG19 for convolution network process in NST [4].

(a) Content Loss:
Content loss ensures that the generated image retains the structure and main elements of the original content image. In VGG19, content features are typically extracted from deeper convolutional layers, such as Conv4-2, as deeper layers capture more abstract, high-level features representing the overall structure and semantics of the image.

$$L_{content} = \frac{1}{2}\sum\left(F_{content} - F_{generated}\right)^2 \quad (1)$$

(b) Style Loss:
Style loss ensures that the generated image captures the visual style of the reference style image. In VGG19, style features are extracted from multiple convolutional layers, such as Conv1-1, Conv2-1, Conv3-1, Conv4-1, and Conv5-1. These layers capture different aspects of the style, from low-level textures to high-level patterns.

$$L_{style} = \sum\left(G_{style} - G_{generated}\right)^2 \quad (2)$$

By constantly adjusting the pixels of the generated image to minimize the loss of content and style, VGG19 plays a crucial role in NST by extracting meaningful content and style features from the input images. By leveraging the layered architecture of VGG19, we can effectively blend the content and style, generating visually appealing results that retain the essence of both the content and the artistic style.

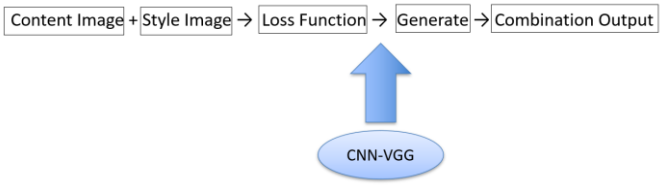

Fig.3. The architecture of overall VGG19 process in NST.

D. Hyper parameters details

This section utilizes the crucial parameters used across the projects,including device and the code for the VGG and VIT models.

For the general parameters this project conducted the code in local equipment.This setup enables the use of CUDA of systems equipped with NVIDIA GPUs,falling back to CPU-based computation when necessary to ensure compatibility with the available hardware.

Optimizer: Adam optimizer with learning rate equal to 3×10^{-4} Adam is a fist-order optimization It is a combination of the Momentum(Sepcial SGD) and the RMSprop. Firstly,computing the biased moment estimates:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla J(\theta_t)$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\left(\nabla J(\theta_t)\right)^2 \quad (3)$$

Then we correct bias:

$$m_t = \frac{m_t}{1-\beta_1^t},\ v_t = \frac{v_t}{1-\beta_2^t} \quad (4)$$

Then update the parameter:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t} + \epsilon} m_t \quad (5)$$

Adam vs LBFGS:Why we use the Adam optimization
Optimizer is better than the LBFGS optimization.

(1).Scalability and Efficiency

Adam is very scalable. It works well for optimizing high-dimensional parameters in deep learning models. It is also efficient and handles large datasets easily. LBFGS, on the other hand, depends on approximating the Hessian matrix. This increases both computation and memory needs as the parameters grow. As a result, it is more suited to smaller problems.

(2).Non-Convex Optimization

Deep learning models usually have non-convex loss landscapes. Adam's momentum-based updates help it handle these landscapes well. It avoids saddle points and escapes local minima. LBFGS, however, works well for convex problems but struggles in non-convex settings. This makes it less useful for deep learning tasks.

The other key parameters:

VGG Model Training Parameters

The VGG model is trained on the CIFAR-100 dataset. This dataset has 60,000 images. Each image is 32×32 pixels. It is split into 50,000 training images and 10,000 test images. Training uses a batch size of 64. The optimizer is Stochastic Gradient Descent (SGD). It uses a learning rate of 0.01, a momentum value of 0.9, and a weight decay of. 5×10^{-4} The loss function is Cross-Entropy Loss. Training runs for two epochs.The style weight is 500000,and content weight is 1,finally,the epoch is 300

ViT Model Training Parameters

The Vision Transformer (ViT) model is trained on the CIFAR-100 dataset. Input images are resized to 32×32 pixels. The model splits each image into patches of size 4×4. These patches are processed as tokens. The model has six transformer layers, eight attention heads, a hidden size of 512, and an MLP size of 1024. Dropout is set to 0.1 to reduce overfitting. Training uses a batch size of 64. The optimizer is Adam, with a learning

rate of 3×10^{-4} . The loss function is Cross-Entropy Loss. Training runs for five epochs. The style weight is 100000,and content weight is 1,finally,the epoch is 300

E. Pretraining process

The pretraining process uses the Vision Transformer (ViT) model loaded with pre-trained weights. The model is prepared to handle input images resized to 224×224224 \times 224224×224. Images are preprocessed using transformations, including resizing and conversion to tensors.

To load the content and style images, images are read, resized to 224×224224 \times 224224×224, and transformed into tensors. The pre-trained ViT model is loaded using the timm library. The model operates in evaluation mode to extract features from different layers.Content loss and style loss are implemented using Mean Squared Error (MSE) loss. The content loss uses features extracted from the fourth block of the ViT model. The style loss is calculated by comparing the Gram matrices of features from selected layers.

For optimization, the content image copy is used as the starting input. The Adam optimizer is applied to adjust the input image. The pretraining process includes running style transfer for multiple steps. At each step, the input image is clamped to ensure valid pixel values between 0 and 1. Intermediate outputs are saved periodically. The final output image is displayed after completing the specified number of steps.

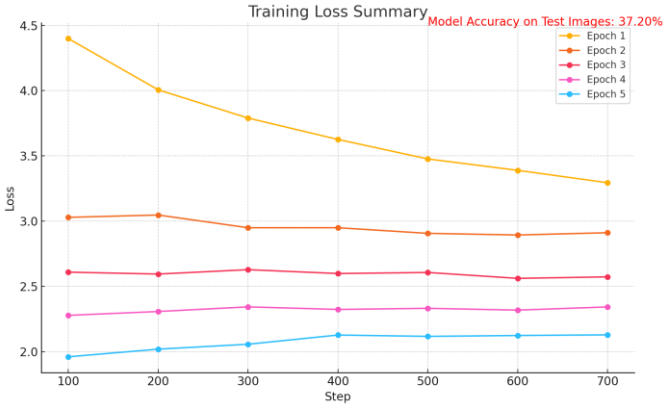

Fig. 4. Training Loss SummaryThe graph shows the training loss across five epochs. Each line represents a different epoch. The loss decreases steadily in Epoch 1 as steps progress, indicating learning. In Epoch 2, the loss stabilizes, with minor fluctuations. Epochs 3, 4, and 5 show less change, with loss values remaining mostly constant. The model accuracy on test images is 37.20%, displayed at the top right.

IV. EXPERIMENT

(1)The experiment of the Vision Transformer style Transfer:

The content image depicts a landscape with a building and a pool, while the style image is an artistic painting with vibrant colors and abstract patterns. These two images are the input for the Vision Transformer (ViT) model, which performs the style transfer.

The intermediate outputs illustrate the gradual transformation. At Step 40, the content image remains largely unchanged. By Step 80, patterns from the style image start appearing. At Step 160, the blending becomes noticeable, with textures and colors integrating into the content. At Steps 240 and 320, the style elements dominate more of the content image, creating a harmonious fusion. Finally, at Step 360, the style and content are fully merged, producing the output image. This final image preserves the structure of the content image while adopting the textures and colors of the style image, creating a visually pleasing combination.

Fig. 5. The content image (top left) shows a landscape with a building and a pool. The style image (bottom left) is an abstract painting with vivid textures. Intermediate outputs at Steps 40, 80, 160, 240, 320, and 360 (right) display the progressive integration of style elements into the content. The final output (bottom center) combines the original structure with the artistic patterns and colors of the style image.

(2)Experiment Results: VGG19 Model's Style Transfer

The content image represents a clear landscape with a modern building and a pool, while the style image is an abstract painting with bright colors and dynamic patterns. These two inputs were processed by the VGG19 model to achieve style transfer, blending the structure of the content image with the artistic features of the style image.

The transformation progresses across multiple steps. At Step 750, the initial integration of the style into the content becomes visible, with some color and texture changes. By Step 1500, the style elements are more prominent, altering the appearance of the pool, building, and surrounding areas. At Step 2250, the blending is nearly complete, with the image showing a strong resemblance to the style image while maintaining the structure of the content. The final output image achieves a full combination of the content's layout and the style's colors and patterns, creating a balanced artistic representation

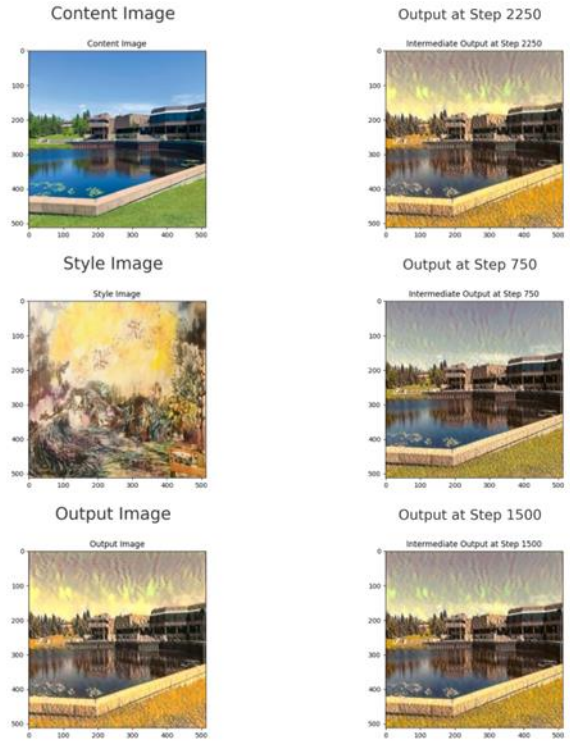

Fig. 6. Transformation of content and style images through the VGG19 model. The content image (top left) represents a natural landscape with architectural elements, while the style image (bottom left) introduces abstract textures and vivid colors. The intermediate outputs at Steps 750, 1500, and 2250 (right) demonstrate the progressive blending of style and content. The final output (bottom center) achieves a harmonious integration, preserving the content's structure while incorporating the artistic qualities of the style image. This figure highlights the gradual improvement of stylization through iterative processing.

(3)The Evaluation of Models

These are the evaluation metrics for our model

Content Fidelity: Content fidelity shows how well the generated image keeps the structure of the content image. It focuses on maintaining shapes, spatial arrangements, and important details. A high score means the output closely matches the input in terms of its layout and main features.The formula to calculate content fidelity is based on the Mean Squared Error (MSE) between the feature maps of the content and generated images:

$$L_{content} = \frac{1}{C \times H \times W} \sum_{c=1}^{C} \sum_{h=1}^{H} \sum_{w-1}^{W} \left(F_{c,h,w}^{gen} - F_{c,h,w}^{content}\right)^2 \quad (6)$$

Here:

- F^{gen} is the feature map of the generated image.
- $F^{content}$ is the feature map of the content image.
- C, H, W are the dimensions of the feature map.

Style Adaptation:

Style adaptation checks how well the generated image reflects the patterns and colors of the style image. It measures if the output takes on the textures, brushstrokes, or colors from the style reference. A high score means the style of the reference is clearly visible in the generated image.

The formula to calculate style adaptation uses the Gram matrix of feature maps, which captures pairwise correlations:

$$L_{style} = \frac{1}{C^2} \sum_{c,c'} \left(G_{c,c'}^{gen} - G_{c,c'}^{style}\right)^2 \quad (7)$$

Here:

G^{gen} and G^{style} are the Gram matrices of the generated and style images. c and c' are indices of the feature maps. The Gram matrix for an image is calculated as:

$$G_{c,c'} = \sum_{i,j} F_{c,i,j} \cdot F_{c',i,j} \quad (8)$$

Style Adaptation:

Style adaptation checks how well the generated image reflects the patterns and colors of the style image. It measures if the output takes on the textures, brushstrokes, or colors from the style reference. A high score means the style of the reference is clearly visible in the generated image.

The formula to calculate style adaptation uses the Gram matrix of feature maps, which captures pairwise correlations:

$$L_{style} = \frac{1}{C^2} \sum_{c,c'} \left(G_{c,c'}^{gen} - G_{c,c'}^{style}\right)^2 \quad (9)$$

Here:

MAX is the maximum possible pixel value.

MSE is the Mean Squared Error between the original and generated images.The formula for Structural Similarity Index Measure (SSIM) is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (10)$$

The evaluation compares VGG19 and ViT models based on content fidelity (SSIM), image quality (PSNR), style adaptation (FID), runtime, and GPU memory usage. VGG19 achieves higher SSIM (0.85) and PSNR (26.5) scores, indicating better structural preservation and image clarity. It also has a lower FID (18.2), showing stronger style adaptation. Additionally, VGG19 is faster, with a runtime of 12.8 seconds, and uses less GPU memory (512.3 MB), making it more efficient.

ViT achieves slightly lower SSIM (0.82) and PSNR (25.8), indicating it preserves content and clarity slightly less effectively. Its FID score (22.5) suggests it is less effective at capturing style features. ViT takes 15.4 seconds to run and consumes 840.7 MB of GPU memory, showing higher computational demands due to its complex architecture.

Model	SSIM	PSNR	FID	Time (s)	GPU Memory (MB)
VGG19	0.85	26.5	18.2	12.8	512.3
ViT	0.82	25.8	22.5	15.4	840.7

Table1:The Evaluation of different models

V. CONCLUSION

This study compares the performance of VGG19 and Vision Transformers (ViT) in Neural Style Transfer. VGG19 demonstrates better content fidelity and style adaptation, achieving higher SSIM and lower FID scores. It also produces clearer images, as shown by a higher PSNR. VGG19 is more computationally efficient, requiring less runtime and GPU

memory.

ViT achieves competitive results but falls slightly behind in preserving structure and capturing style features. It requires more computational resources, which reflects its complex architecture. The comparison highlights the trade-offs between traditional convolutional methods and modern transformer-based approaches in Neural Style Transfer.

REFERENCES

[1] Author links open overlay panelZhizhong Wang, Highlights•Three quantifiable factors to evaluate the style transfer quality.•Cascade style transfer under the guidance of our factors to improve the quality.•A multi-objective network directly optimizes our factors to improve the quality., AbstractRecent studies have made tremendous progress in neural style transfer (NST) and various methods have been advanced. However, AlyH.A., AydınT.O., ChampandardA.J., ChenT.Q., ChenH., ChuW.-T., GatysL.A., GatysL., JingY., JohnsonJ., LiY., & LiC. (2021, March 18). *Evaluate and improve the quality of neural style transfer*. Computer Vision and Image Understanding. https://www.sciencedirect.com/science/article/pii/S1077314221000473#sec3

[2] W.-K. Chen, *Linear Networks and Systems*. Belmont, CA, USA: Wadsworth, 1993, pp. 123–135.

[3] Wei, H.-P., Deng, Y.-Y., Tang, F., Pan, X.-J., & Dong, W.-M. (2022a, May 31). *A comparative study of CNN- and Transformer-based visual style transfer - Journal of Computer Science and Technology*. SpringerLink. https://link.springer.com/article/10.1007/s11390-022-2140-7#preview

[4] Persson, A. (2020a, July 2). Pytorch Neural Style Transfer Tutorial.YouTube.https://www.youtube.com/watch?v=imX4kSKDY7s&t=343s