# Unpaired Image Translation

**Danna Gurari**
University of Colorado Boulder
Fall 2024

# Review

- Last lecture topic:
  - Foundation Models
  - Textual Prompting & Zero-shot Learning
  - Visual Prompting & In-context Few-shot Learning
  - Prompt Tuning
  - Discussion
- Assignments (Canvas)
  - Project outline due earlier today
  - Reading assignments due before each class meeting until Fall break
- Questions?

# Unpaired Image Translation: Today’s Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches

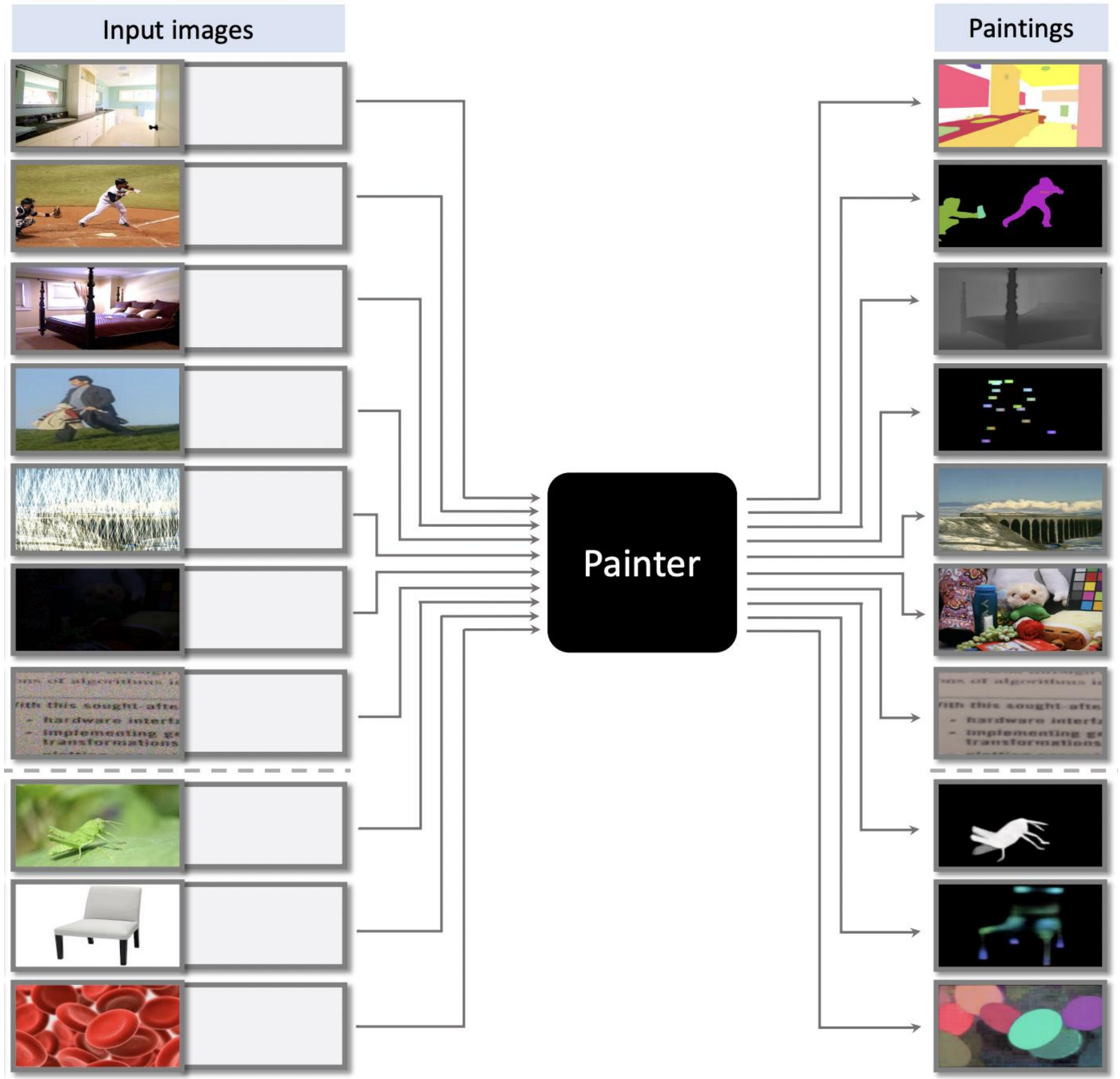
# Unpaired Image Translation: Today's Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches

# Image-to-Image Translation

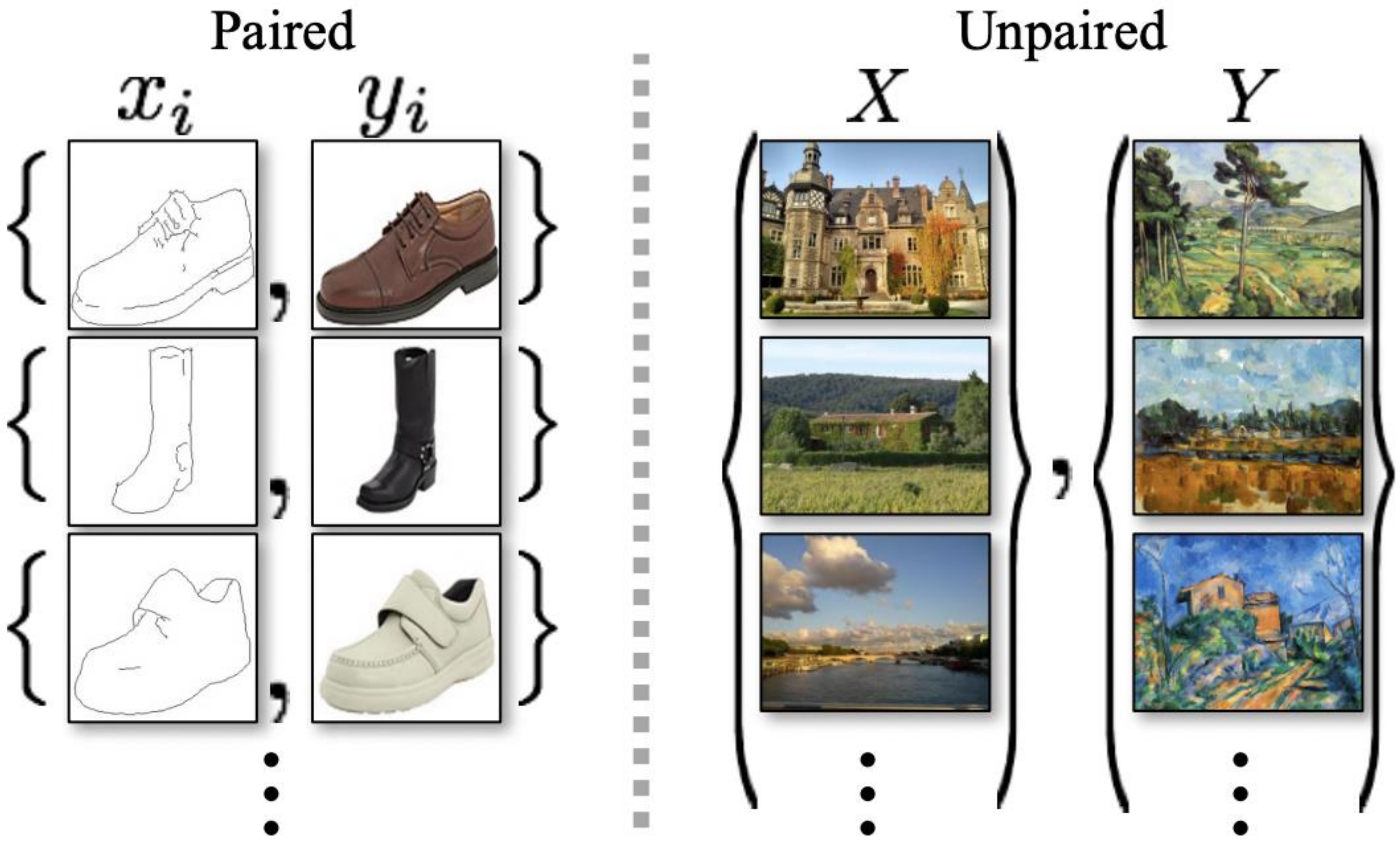
- Learn mapping between two image representations; e.g.,

- Today's scope: learn the mappings without paired training data (i.e., input output examples)




Wang et al. Images Speak in Images: A Generalist Painter for In-Context Visual Learning. CVPR 2023

# Paired vs Unpaired Image Translation

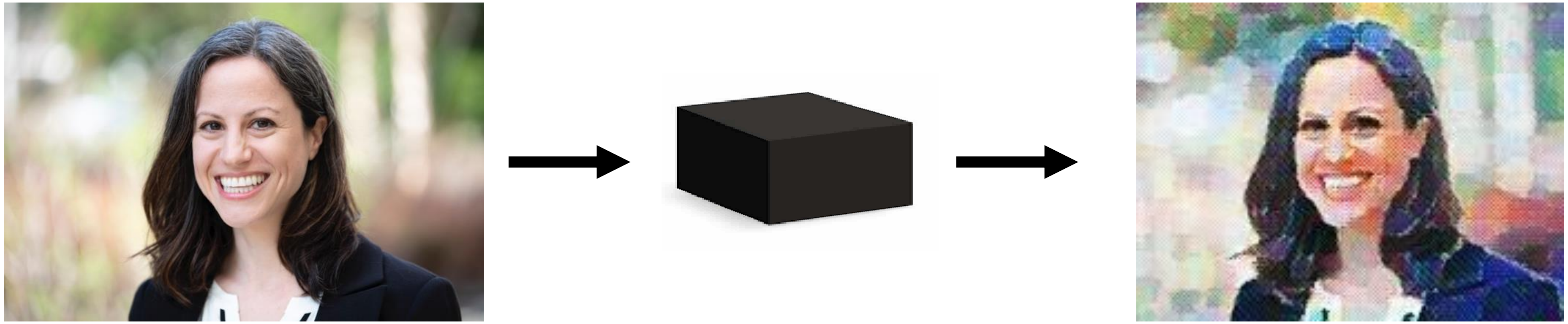



No mapping of inputs (X) to outputs (Y)

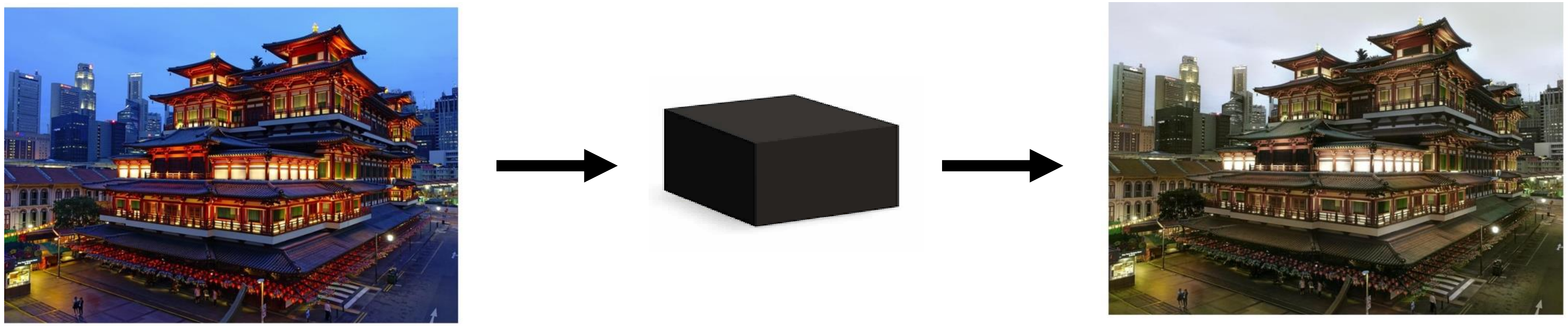
Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV 2017

# An Unpaired Image Translation Task: Transform **Content** of Image into a New **Style**

Artistic:



Photorealistic:



# An Unpaired Image Translation Task: Transform **Content** of Image into a New **Style**

How would you define “content”?


Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016.

# An Unpaired Image Translation Task: Transform **Content** of Image into a New **Style**

How would you define “style”?

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016.

# Key Challenges

- How to computationally isolate the content of the content image?
- How to computationally isolate the style of the style image?
- How to blend the content and style?
- How to support any arbitrary style?
- How to do all these fast?

# Style Transfer: Today's Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches

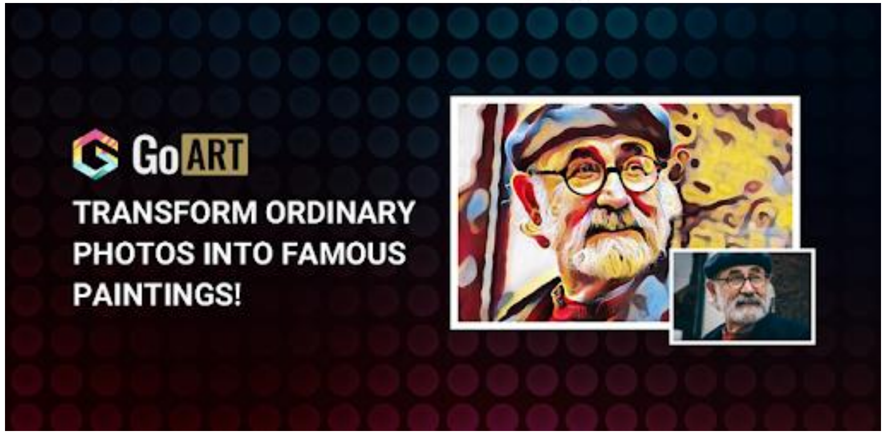
# Entertainment (Mobile Phone Applications)

Browser demo: https://reiinakano.com/arbitrary-image-stylization-tfjs/

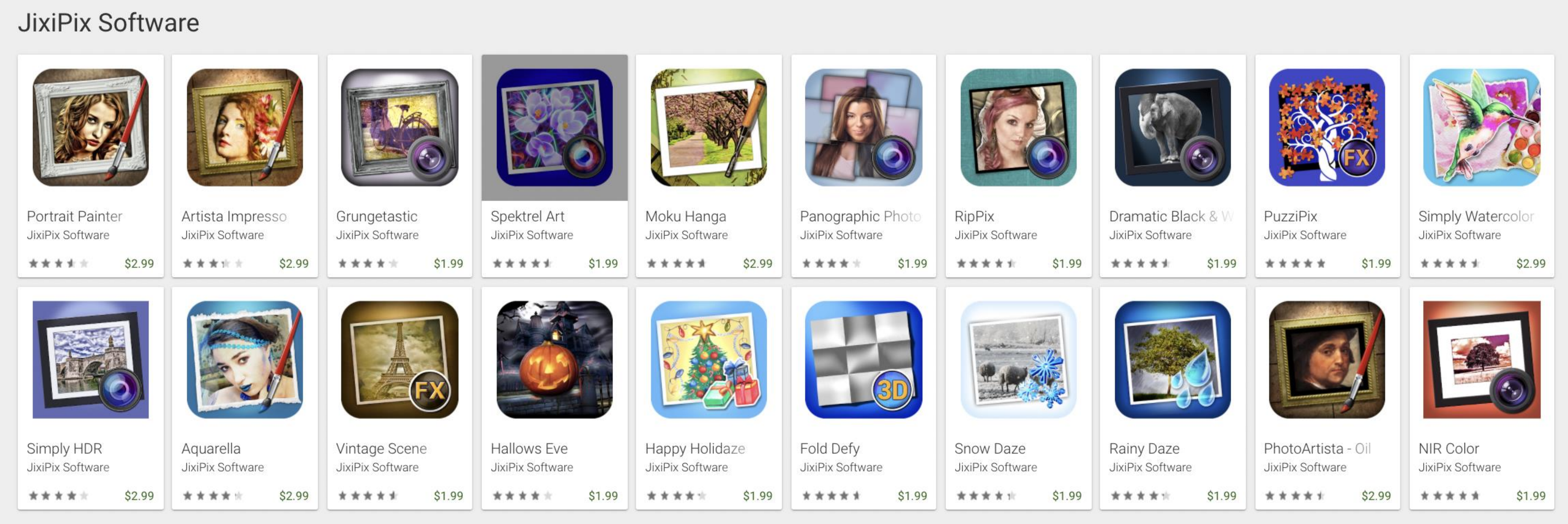
# Entertainment (Mobile Phone Applications)


ARTIFY

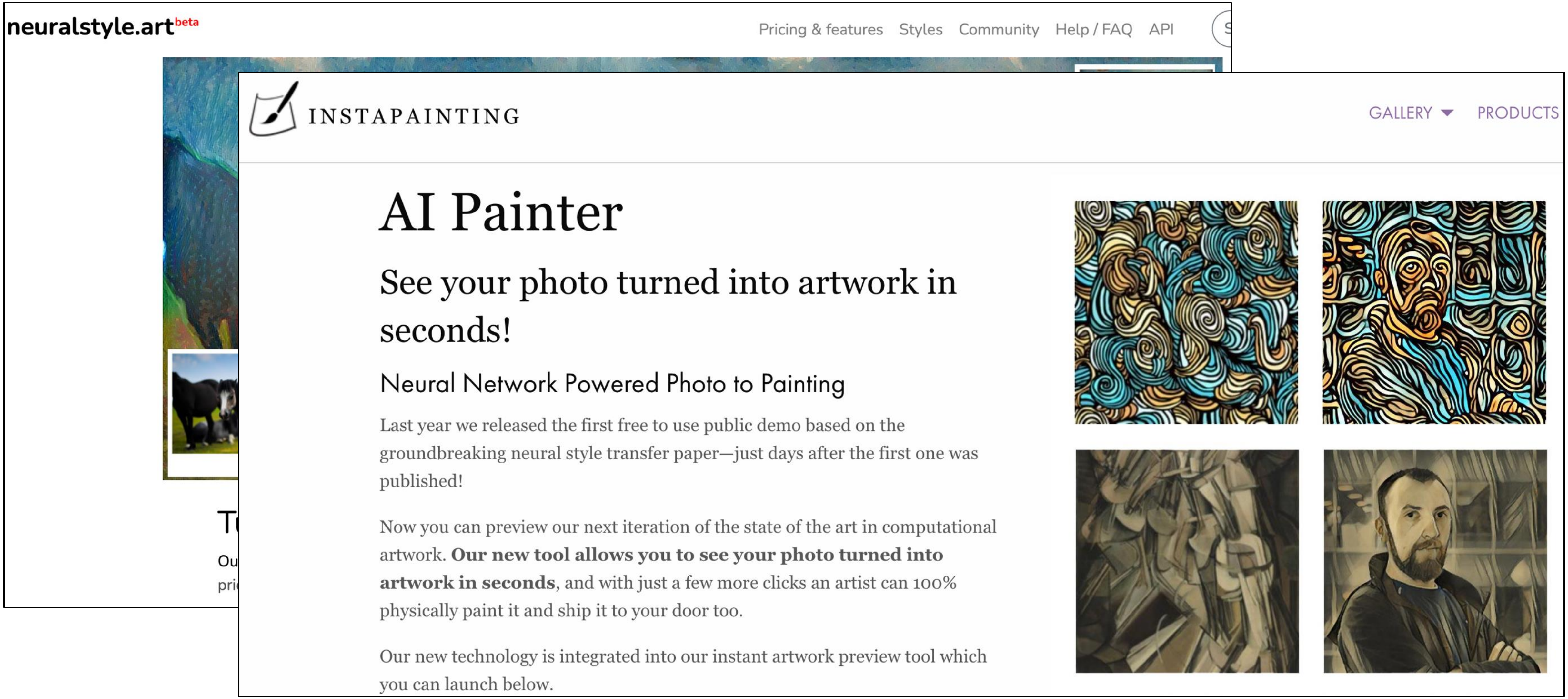

PicsArt




# Entertainment (Mobile Phone Applications)



# Commercial Art



# Virtual and Augmented Reality


Demo: https://youtu.be/Rz4J3T1uYYo

# Virtual and Augmented Reality

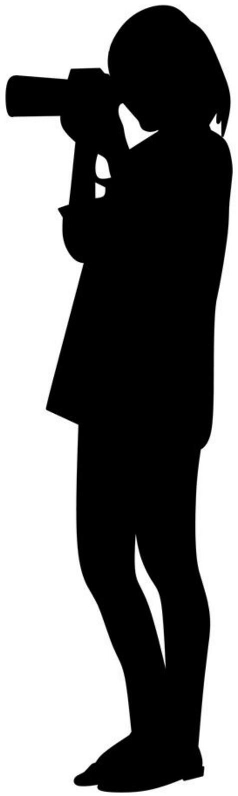
Demo: https://www.youtube.com/watch?v=pkgMUfNeUCQ

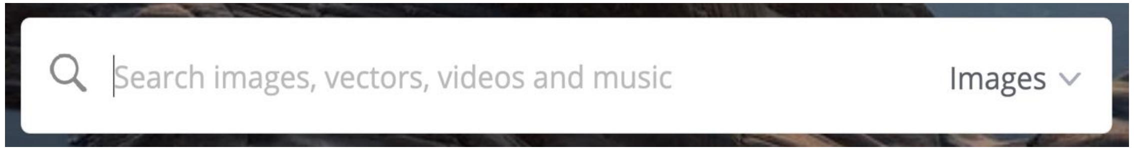
# Gaming (e.g., Stadia from Google)


Demo: https://www.youtube.com/watch?v=yF1bZiH-wJQ

# Improve Messaging via Visual Content

- Marketing
- Artwork
- Presentations
- Blogs
- Websites






shutterstock
pixabay
Pexels
St Adobe Stock
BIGSTOCK™
iStock by Getty Images
dreamstime®
Google Images
Unsplash Photos for everyone

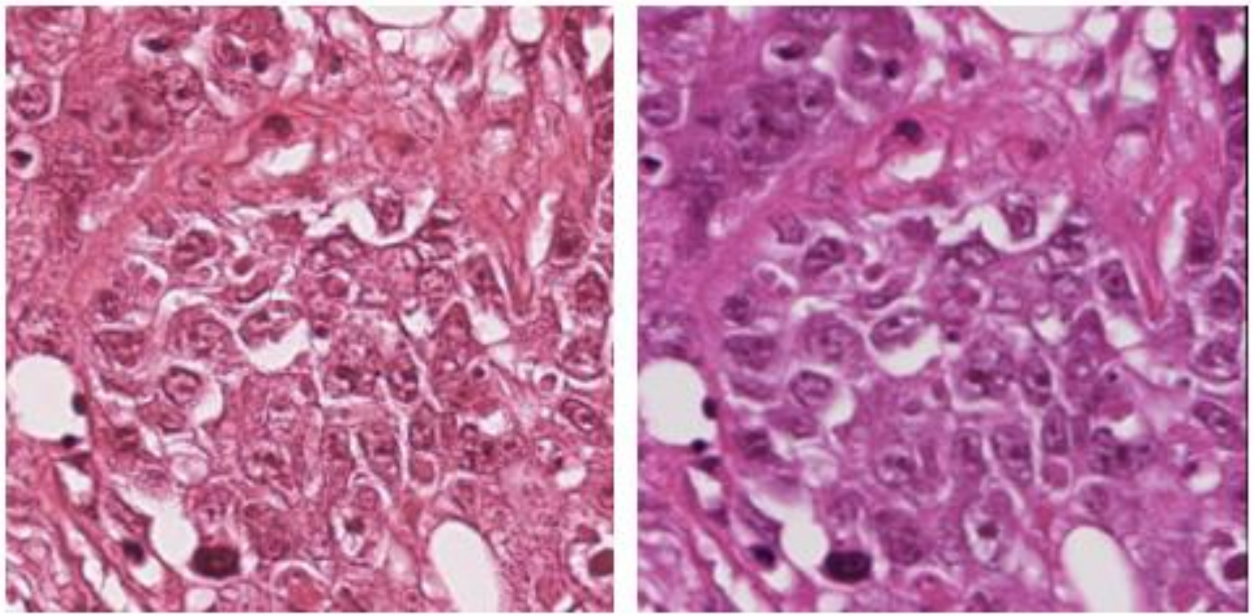
Potential sources:

Photographer (self or hired)

Stock photos

# Improve Quality of Data for AI Analysis

Breast cancer classification



(a) Source (b) Target

Shaban et al. ISBI 2019

What are other possible applications for style transfer?

# Style Transfer: Today’s Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches

# Neural Style Transfer (NST): Addresses…

- How to computationally isolate the content of the content image?
- How to computationally isolate the style of the style image?
- How to blend the content and style?
- How to support any arbitrary style?
- How to do all these fast?

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST): Key Insight

"The representations of content and style in the Convolutional Neural Network are well separable."

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST)

Approach: iteratively modify a random image guided by the content image and style image

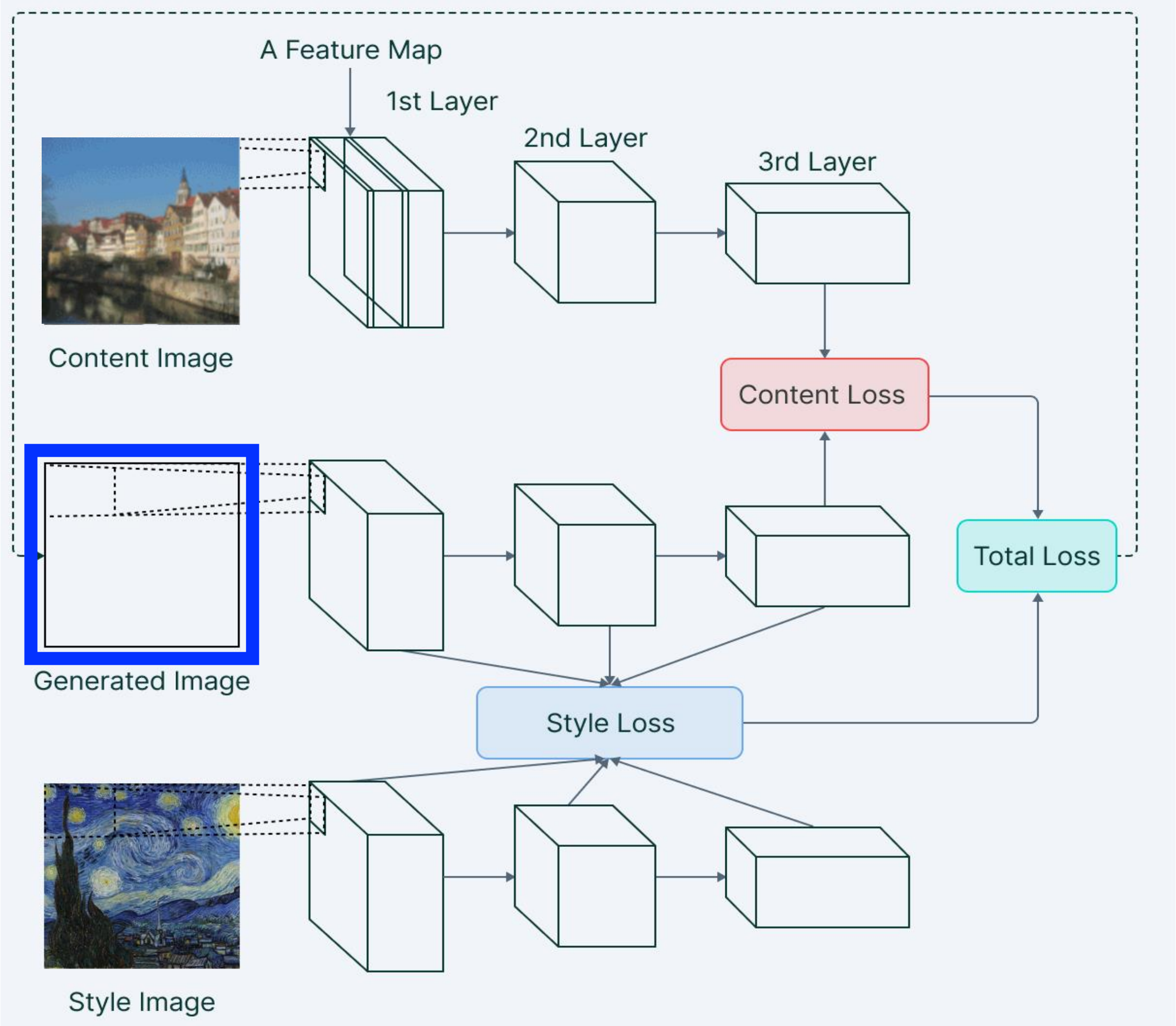



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST)

Approach: iteratively modify a random image guided by the content image and style image

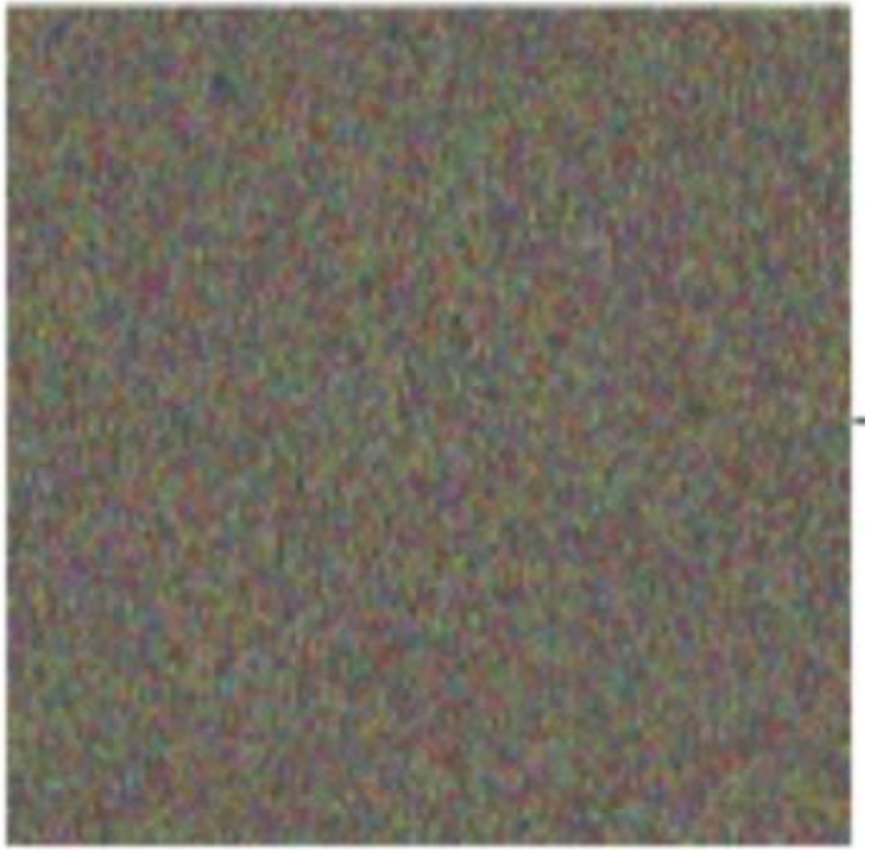


Figure Source: https://towardsdatascience.com/a-brief-introduction-to-neural-style-transfer-d05d0403901d

# Neural Style Transfer (NST)

Approach: iteratively modify a random image guided by the content image and style image

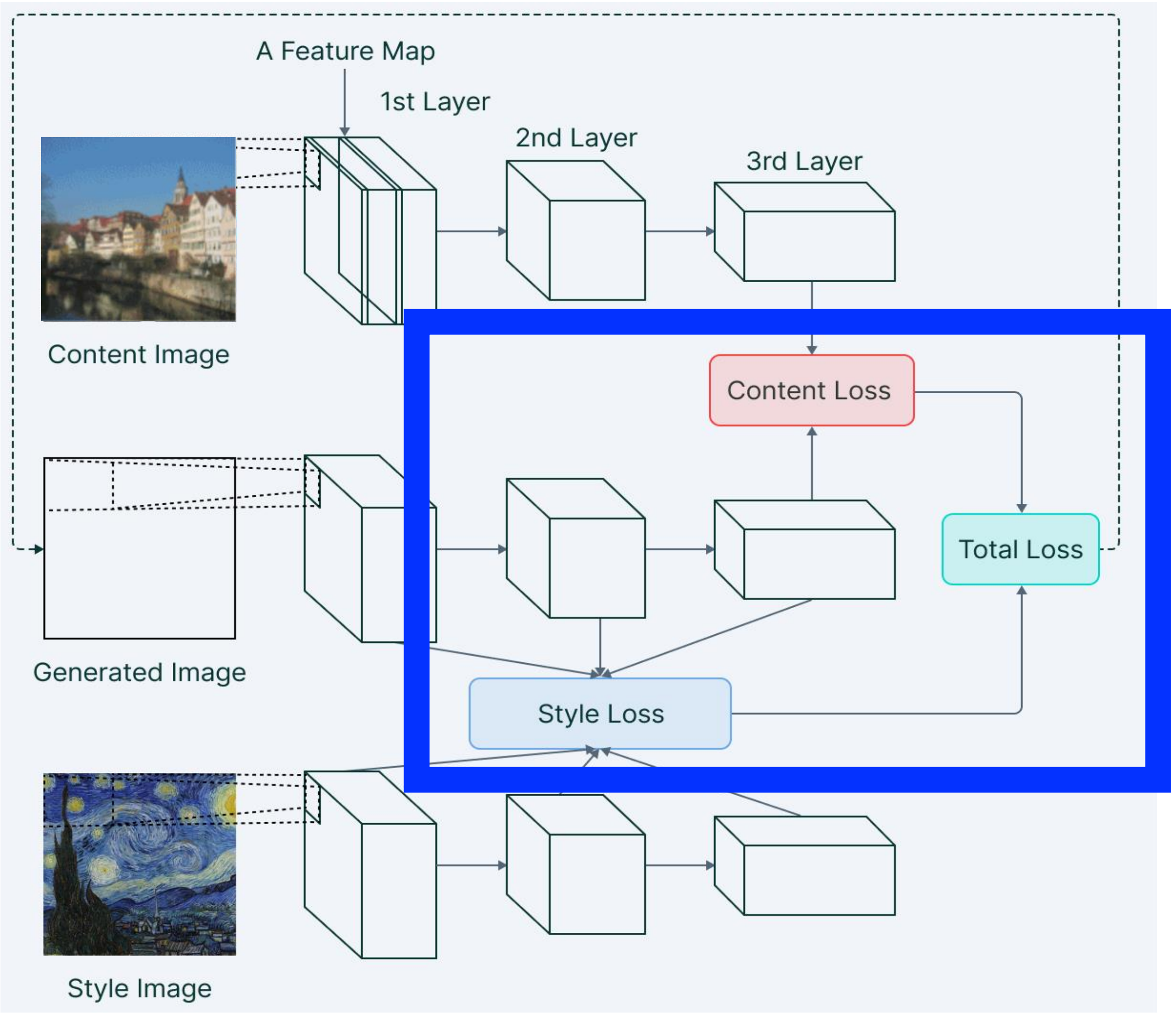



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST)

Approach: iteratively modify a random image guided by the content image and style image

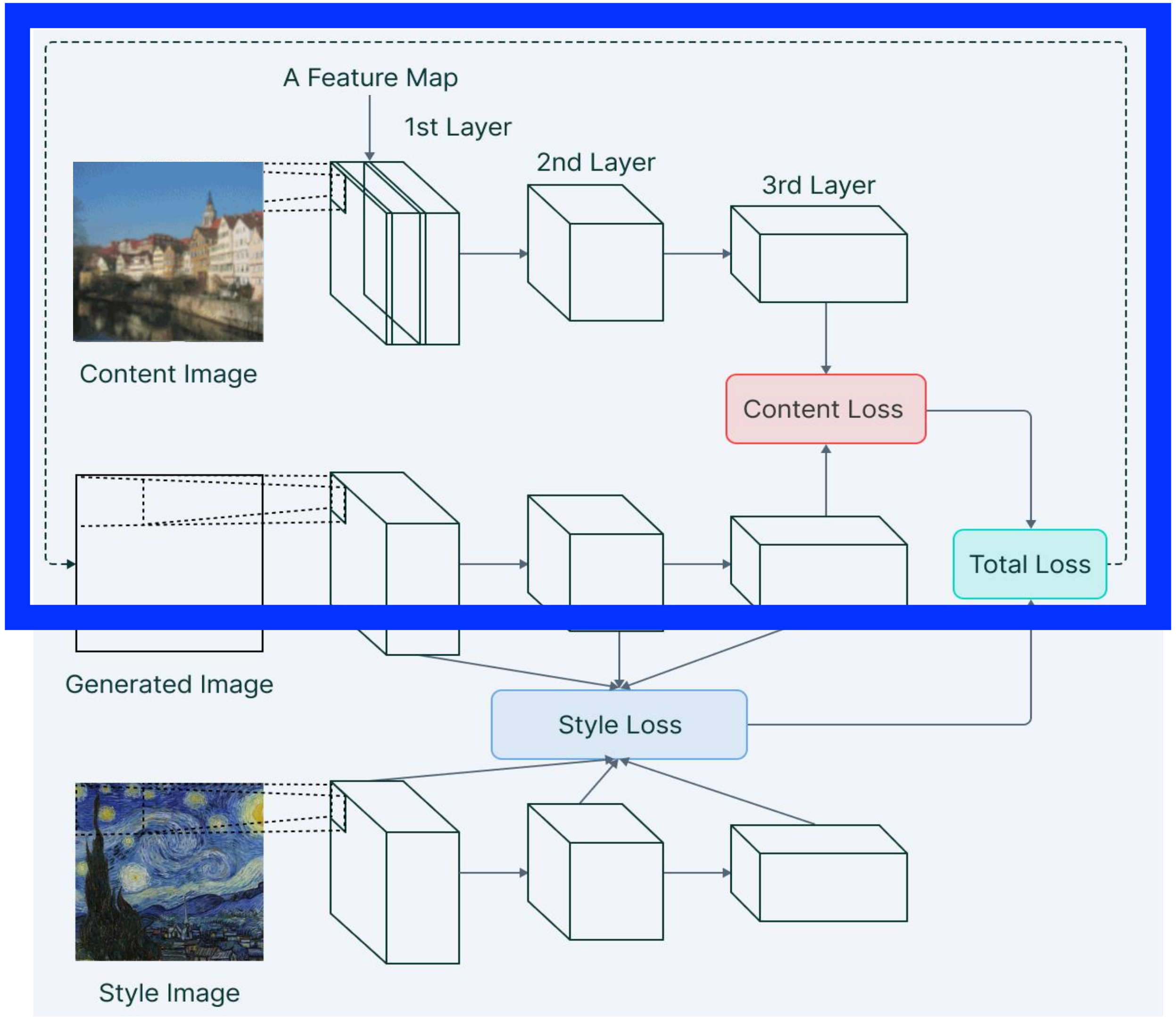



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST)

- How to computationally isolate the content of an image?
  - Recall, what CNNs typically learn:

Content representation: feature maps often show spatial structure without texture/style

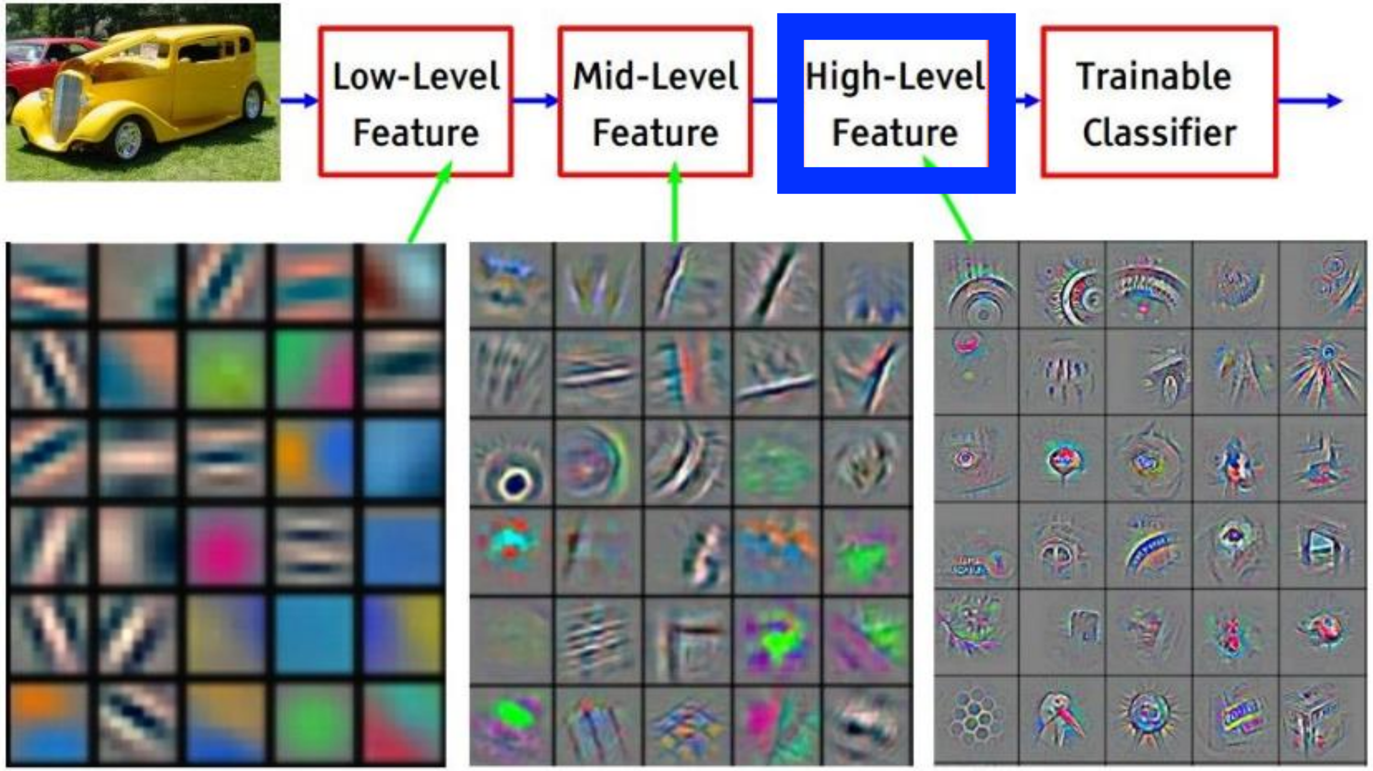



Figure Credit: Yann LeCun

# Neural Style Transfer (NST)

Iteratively adjust the generated image until its high level features match the high level features of the content image

Layer of the network

$$\mathcal{L}_{\text{content}}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{ij}\left(F^l_{ij} - P^l_{ij}\right)^2$$

Index of feature map generated by the *i*-th filter in layer *l* of the network

Position to look at in the feature map

Approach: iteratively modify a random image guided by the content image and style image

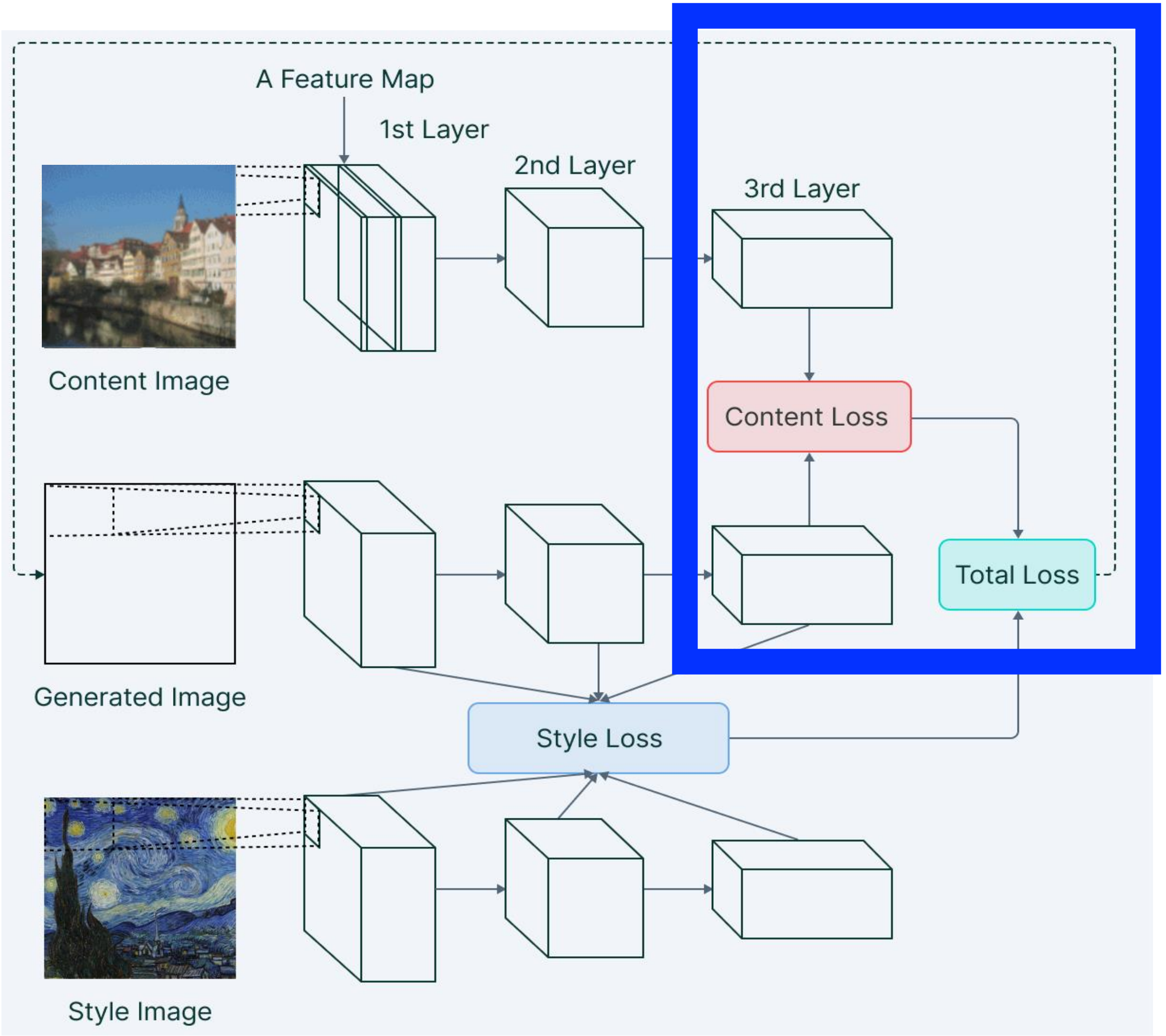



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST)

Iteratively adjust the generated image until its feature correlations match the feature correlations of the style image

Approach: iteratively modify a random image guided by the content image and style image

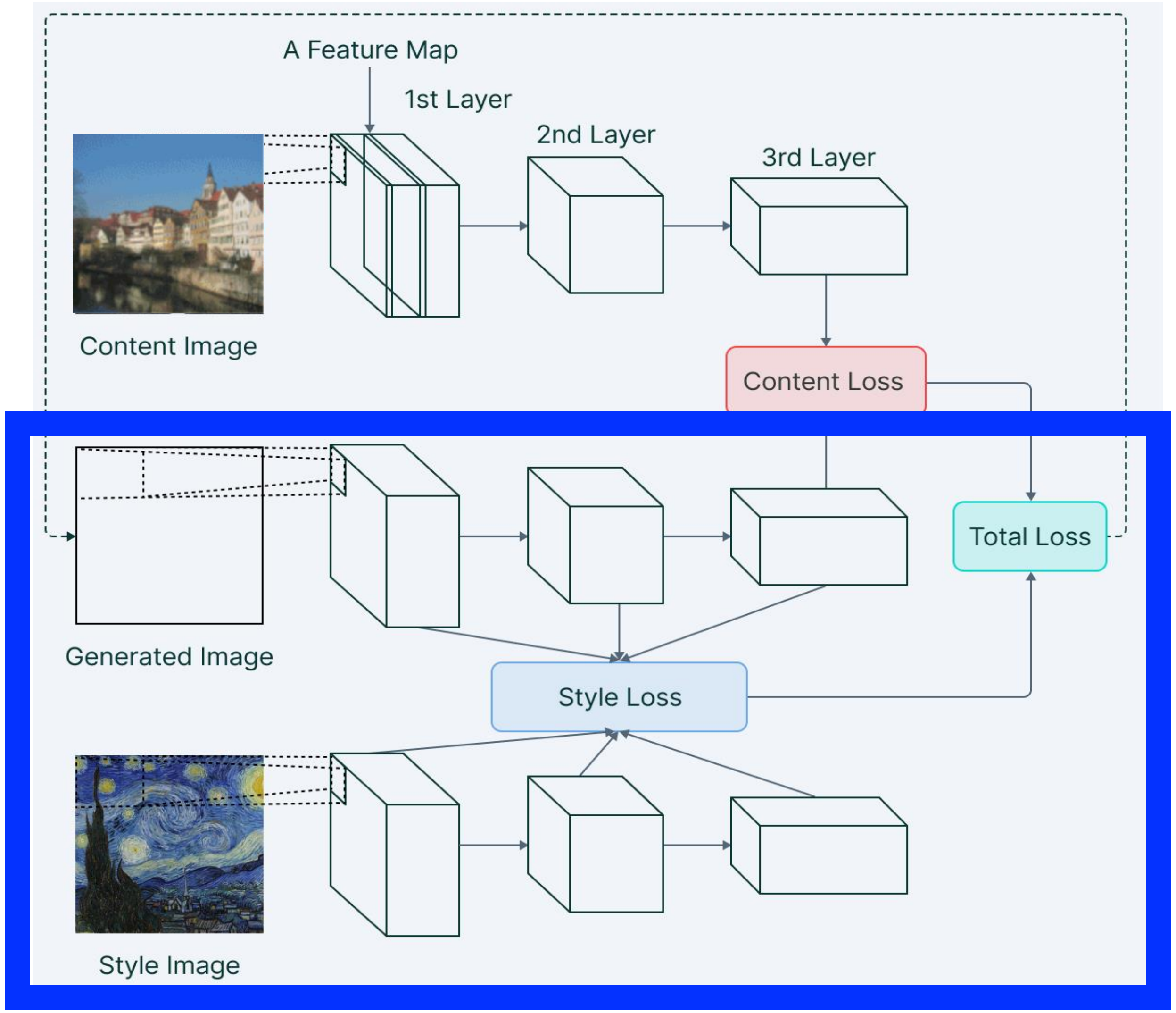



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST)

Iteratively adjust the generated image until its feature correlations match the feature correlations of the style image

Approach: iteratively modify a random image guided by the content image and style image

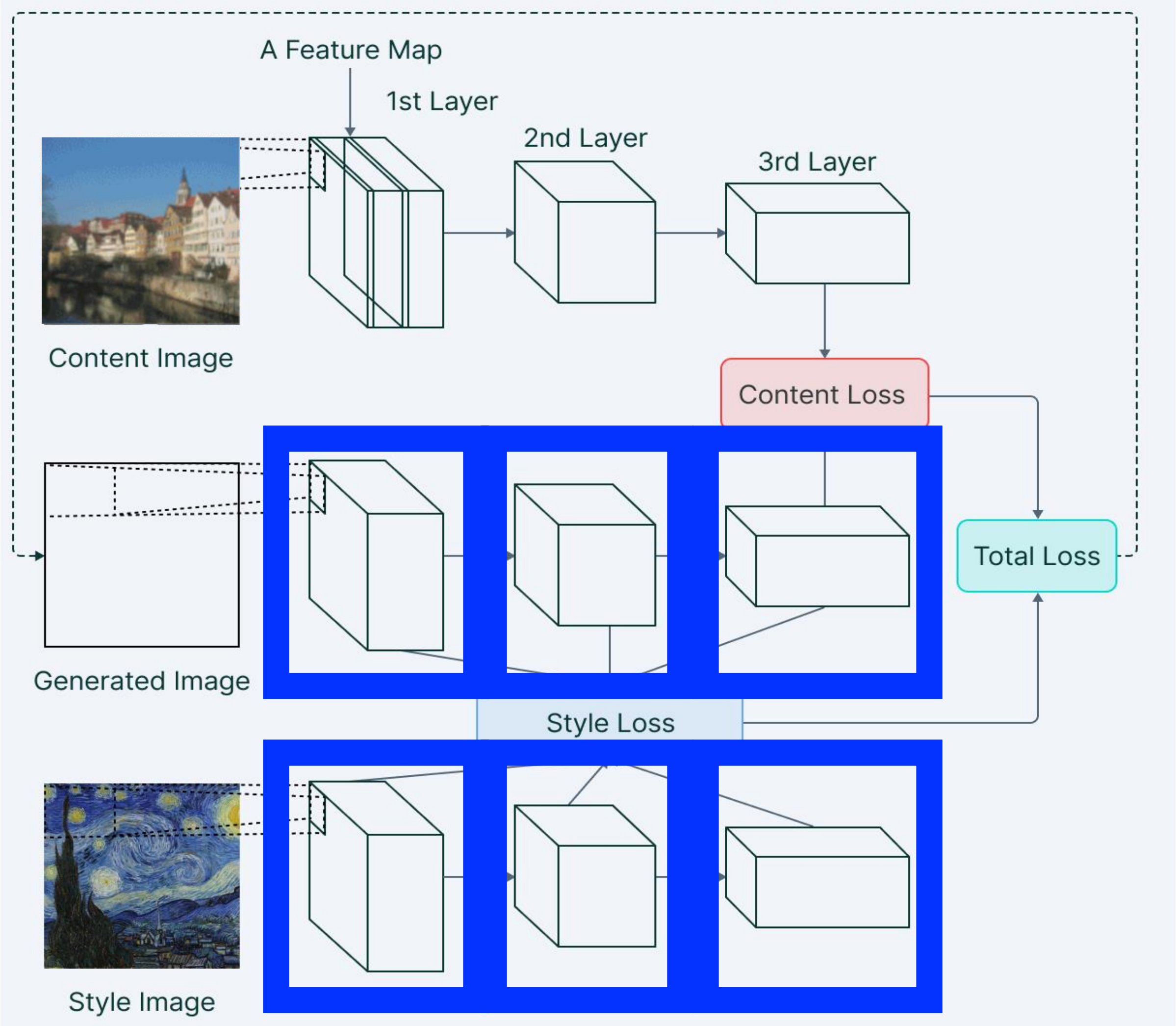
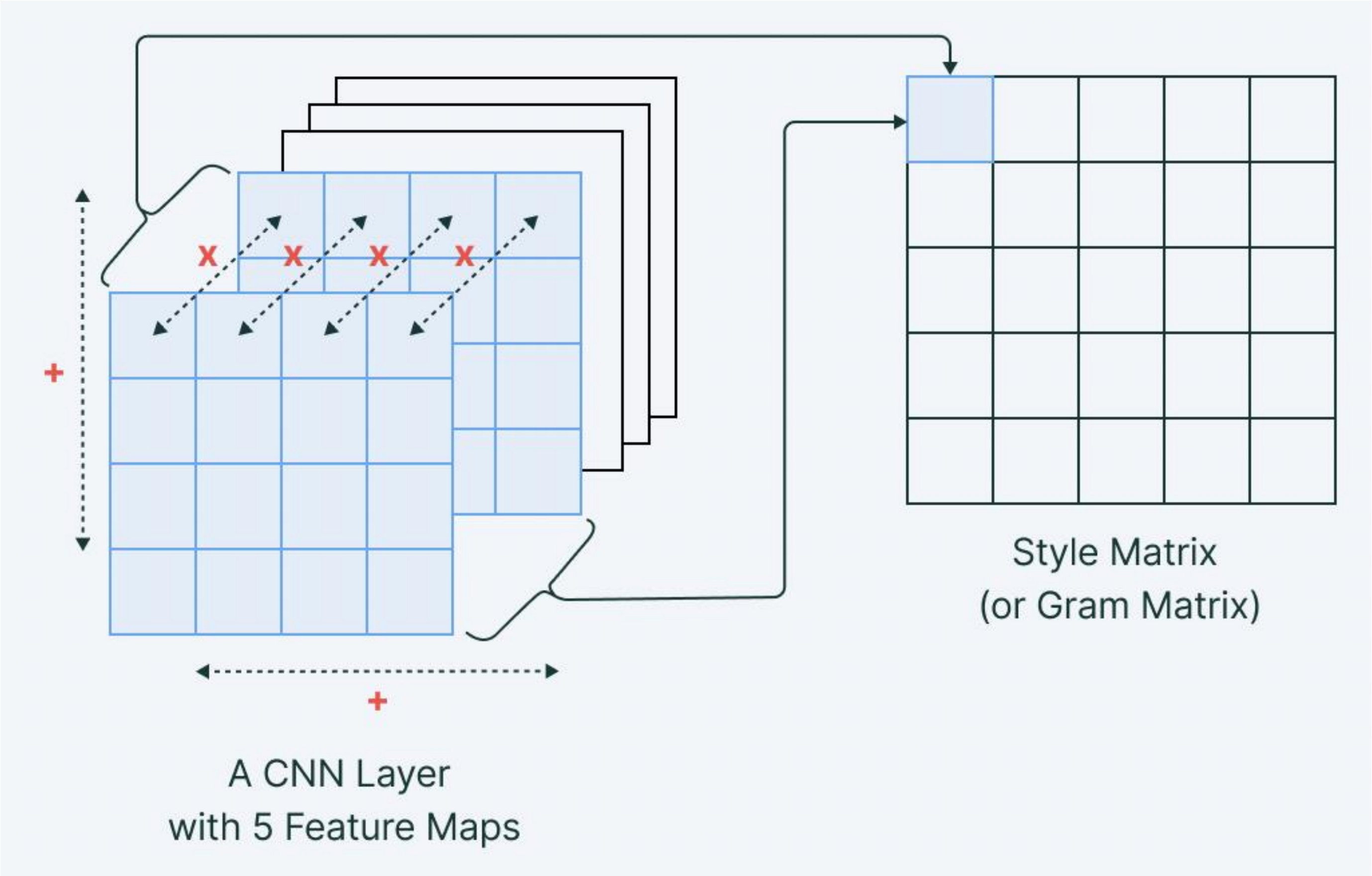



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST): Gram Matrix Used to Represent an Image's Style



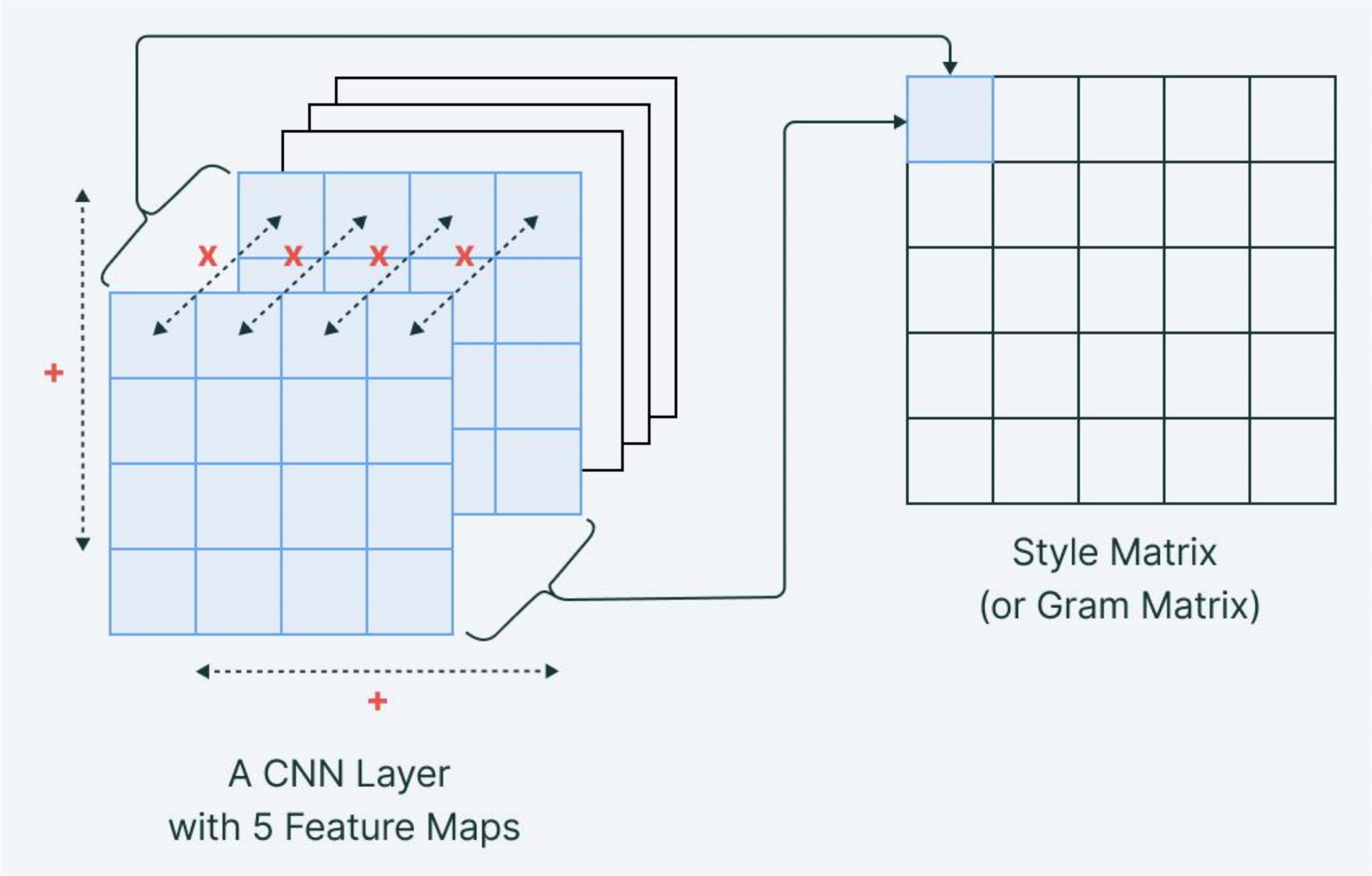

For a layer, correlation computed between its features maps (i.e., gram matrix)

- each 2d map flattened into 1d (which removes structure info)

- dot product computed for each 1d vector with itself and others (larger values indicate greater feature co-occurrence)

Figure Source: https://www.v7labs.com/blog/neural-style-transfer

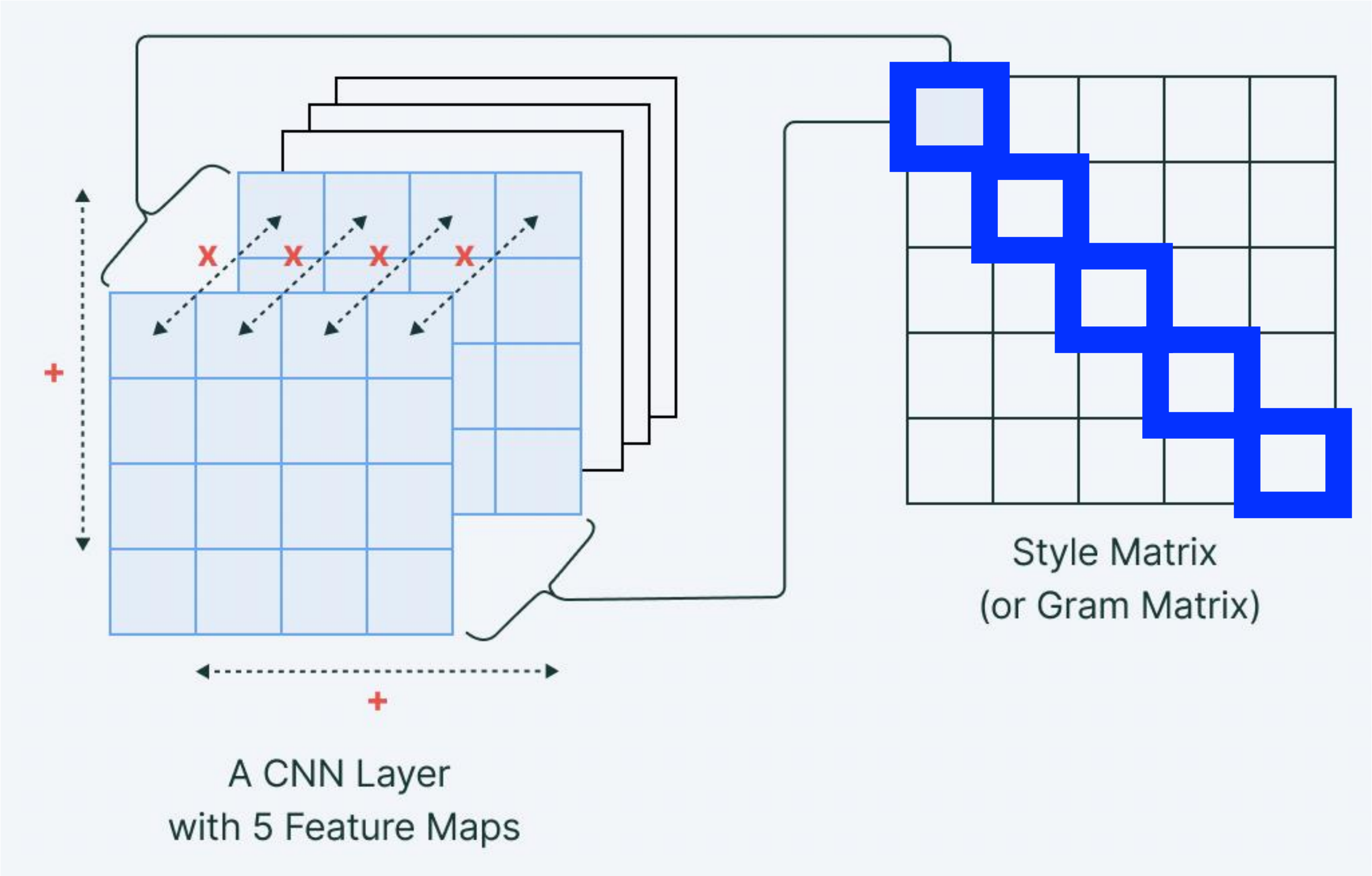
# Neural Style Transfer (NST): Gram Matrix Used to Represent an Image's Style




We know we start with *N* feature maps each containing *M* values. What will be the dimension of the Gram matrix?
- N x N

Figure Source: https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST): Gram Matrix Used to Represent an Image’s Style




What should be the values on the diagonal of the Gram matrix?
- 1 (reflects perfect match between a feature map and itself)

Figure Source: https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST)

Iteratively adjust the generated image until its feature correlations match the feature correlations of the style image

Loss for 1 layer:

Layer of the network

$$E_{\boxed{l}} = \frac{1}{\boxed{4N_l^2 M_l^2}} \sum_{i,j} \left(\boxed{G_{ij}^l} - \boxed{A_{ij}^l}\right)^2$$

Number of feature maps x each map's size in layer *l*

Gram matrices

Approach: iteratively modify a random image guided by the content image and style image

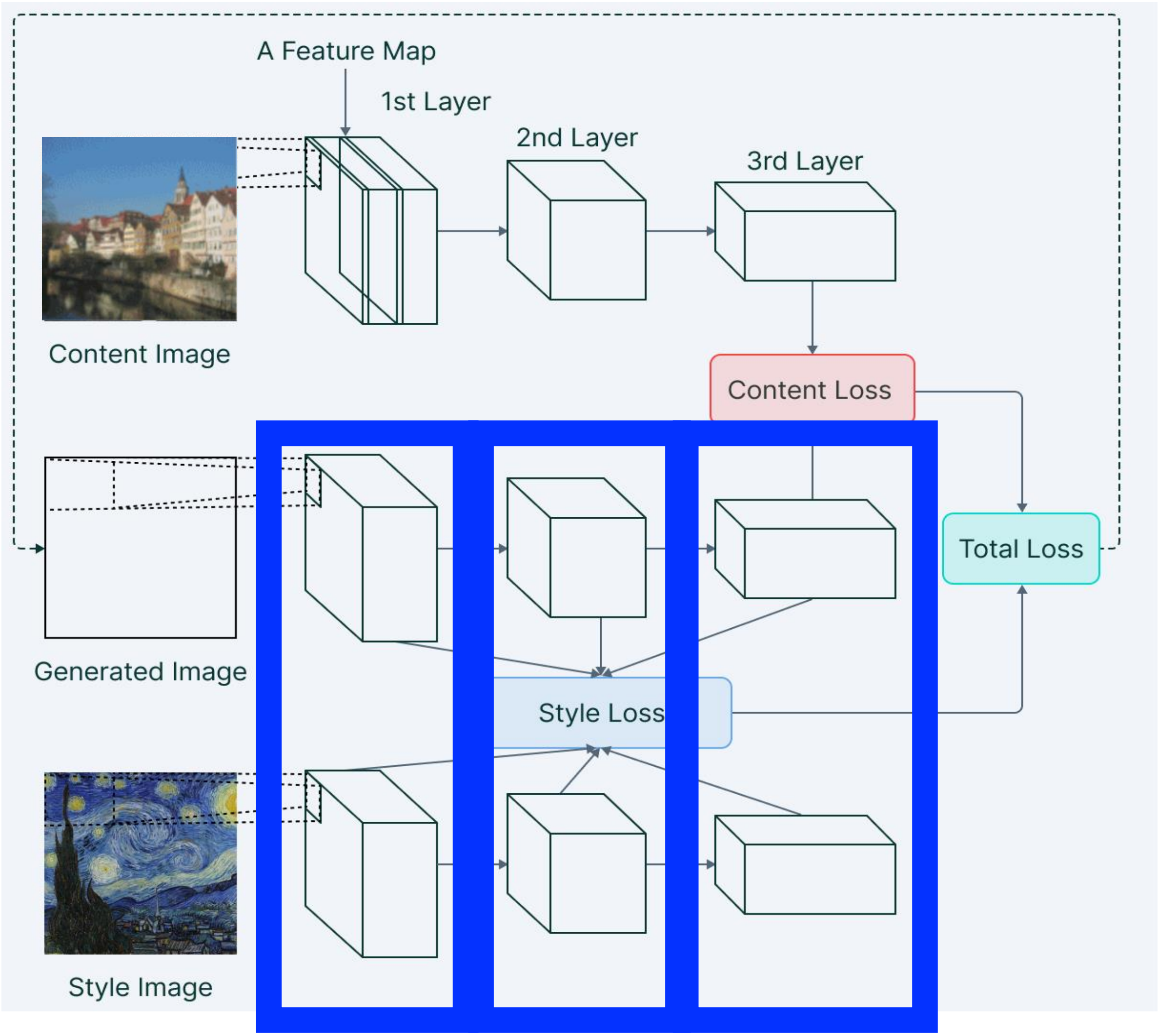



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST)

- Which layers to use to computationally isolate an image's style?
  - Recall, what CNNs typically learn:

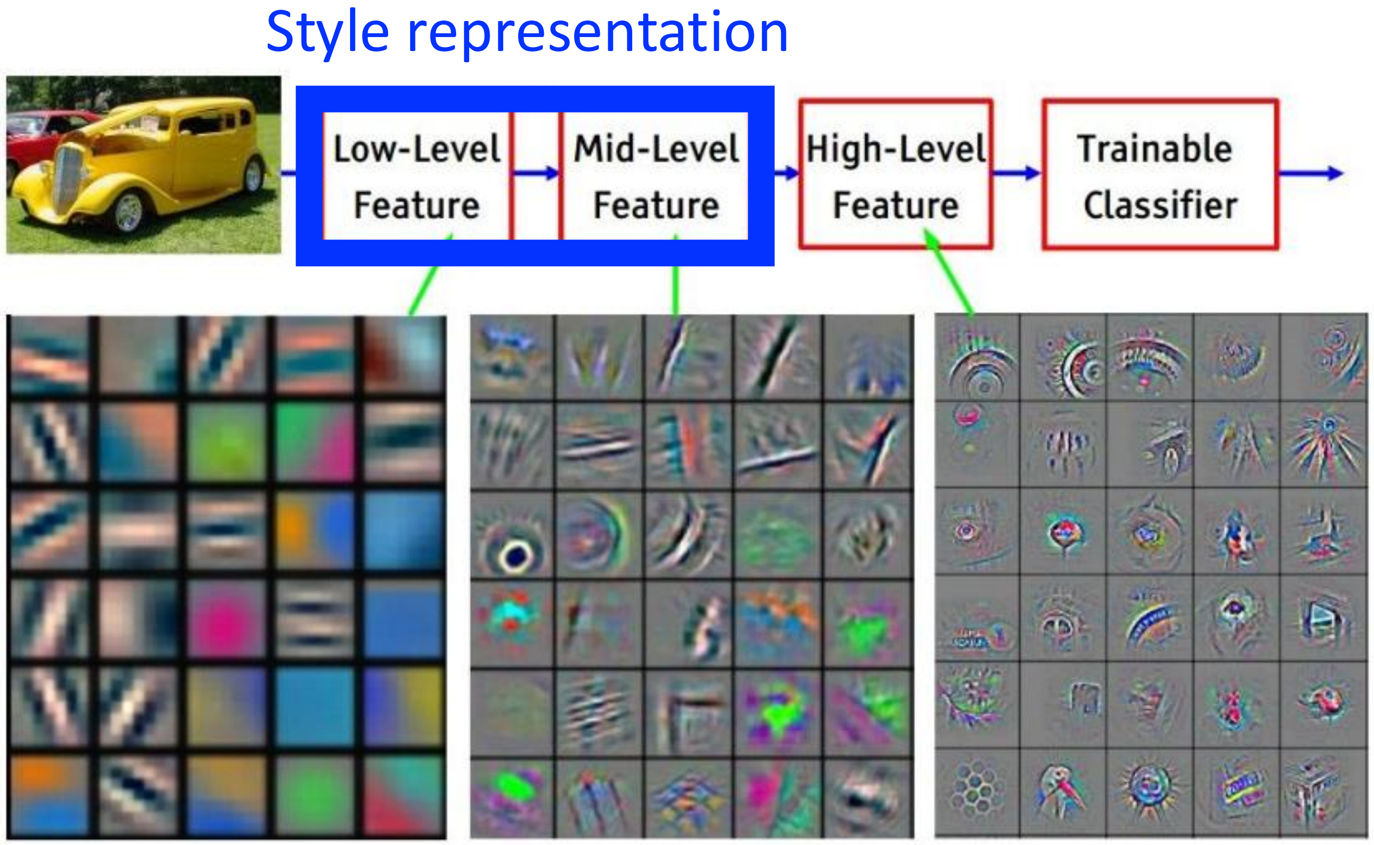



Figure Credit: Yann LeCun

# Neural Style Transfer (NST)

Iteratively adjust the generated image until its feature correlations match the feature correlations of the style image

Total loss is the weighted sum of correlation differences across all layers

$$\mathcal{L}_{\text{style}}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l,$$

Approach: iteratively modify a random image guided by the content image and style image

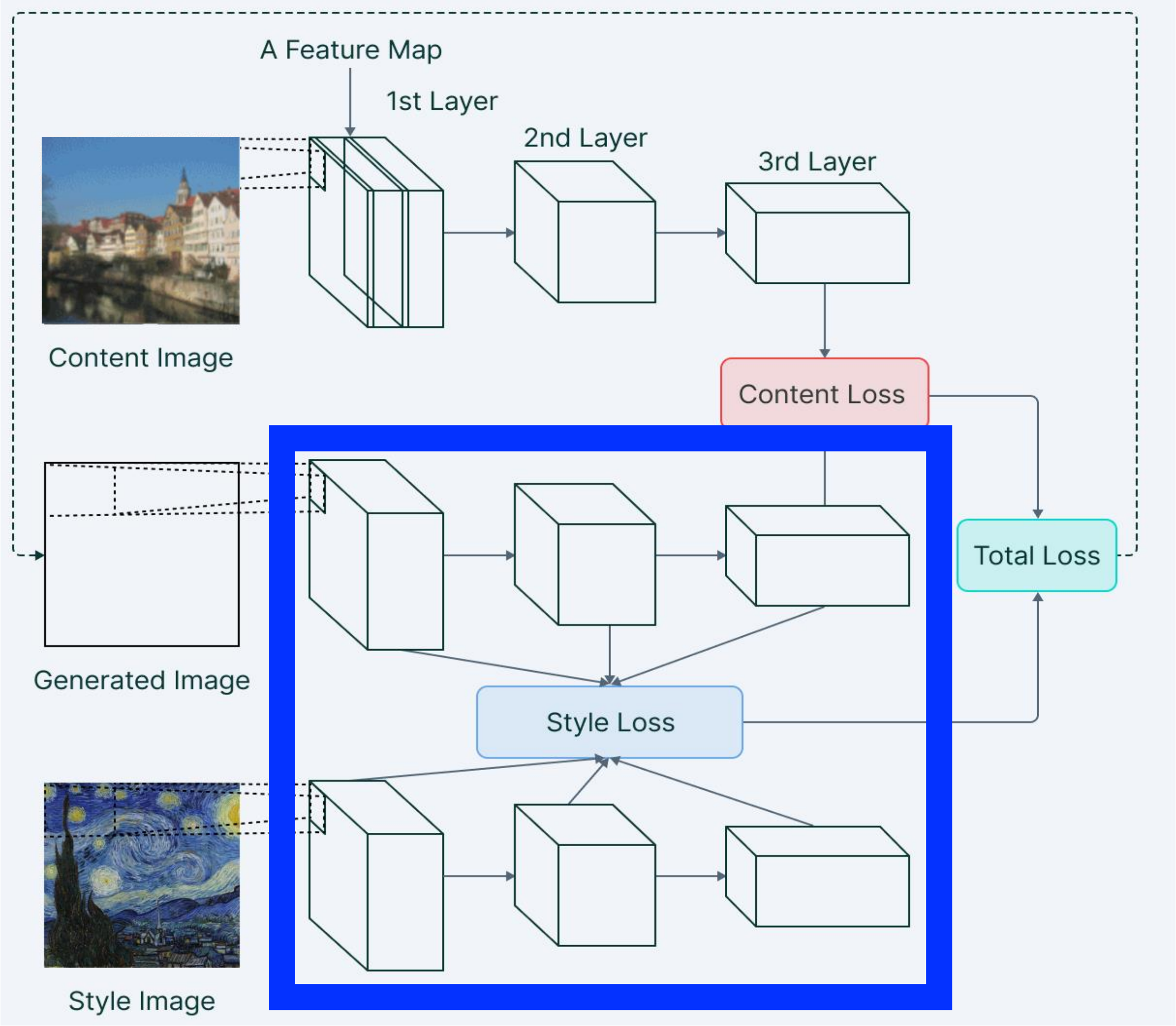
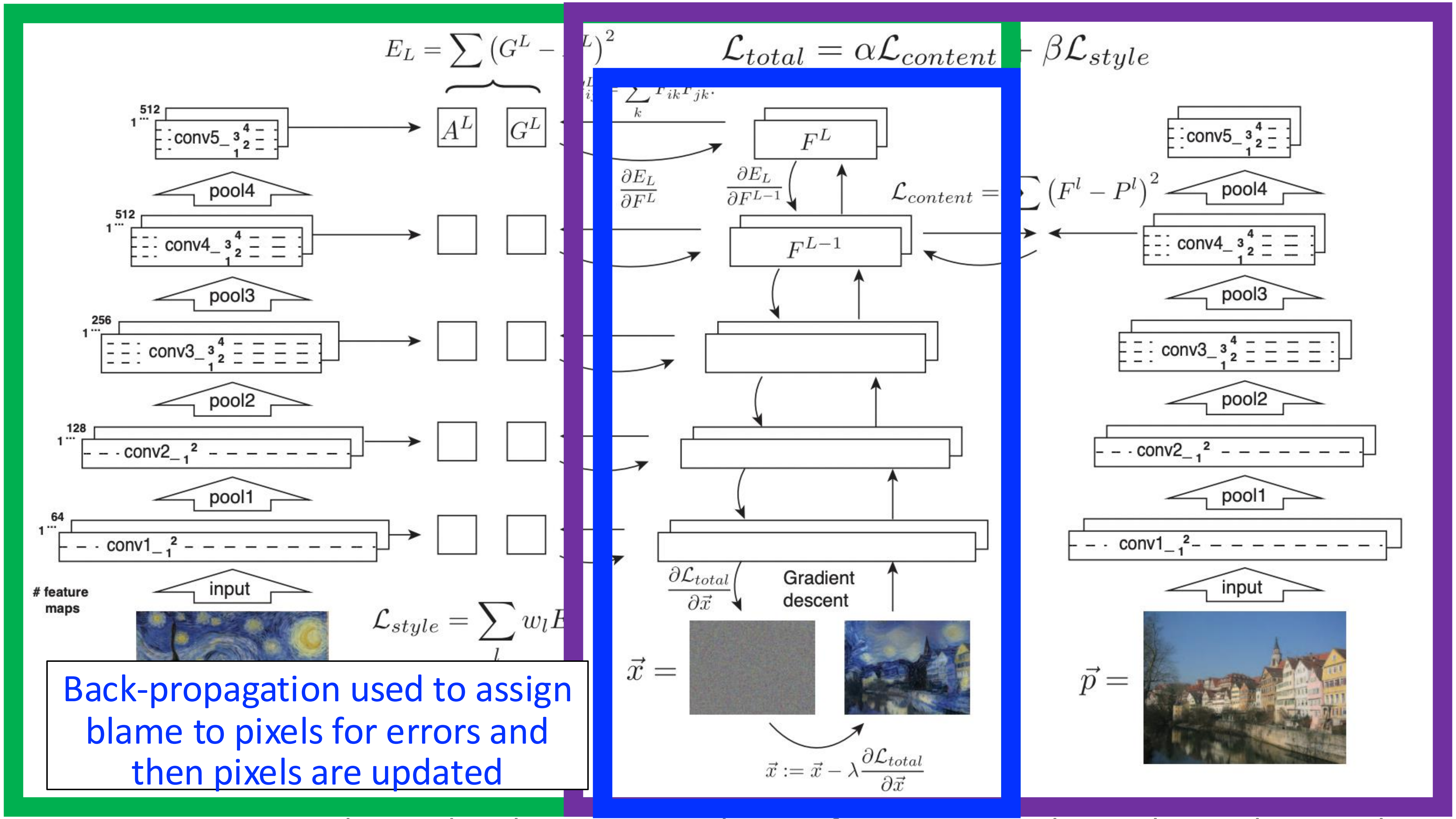



Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Neural Style Transfer (NST): Algorithm

Compute content loss based on feature maps from 1 layer

Compute style loss based on feature maps from 5 layers



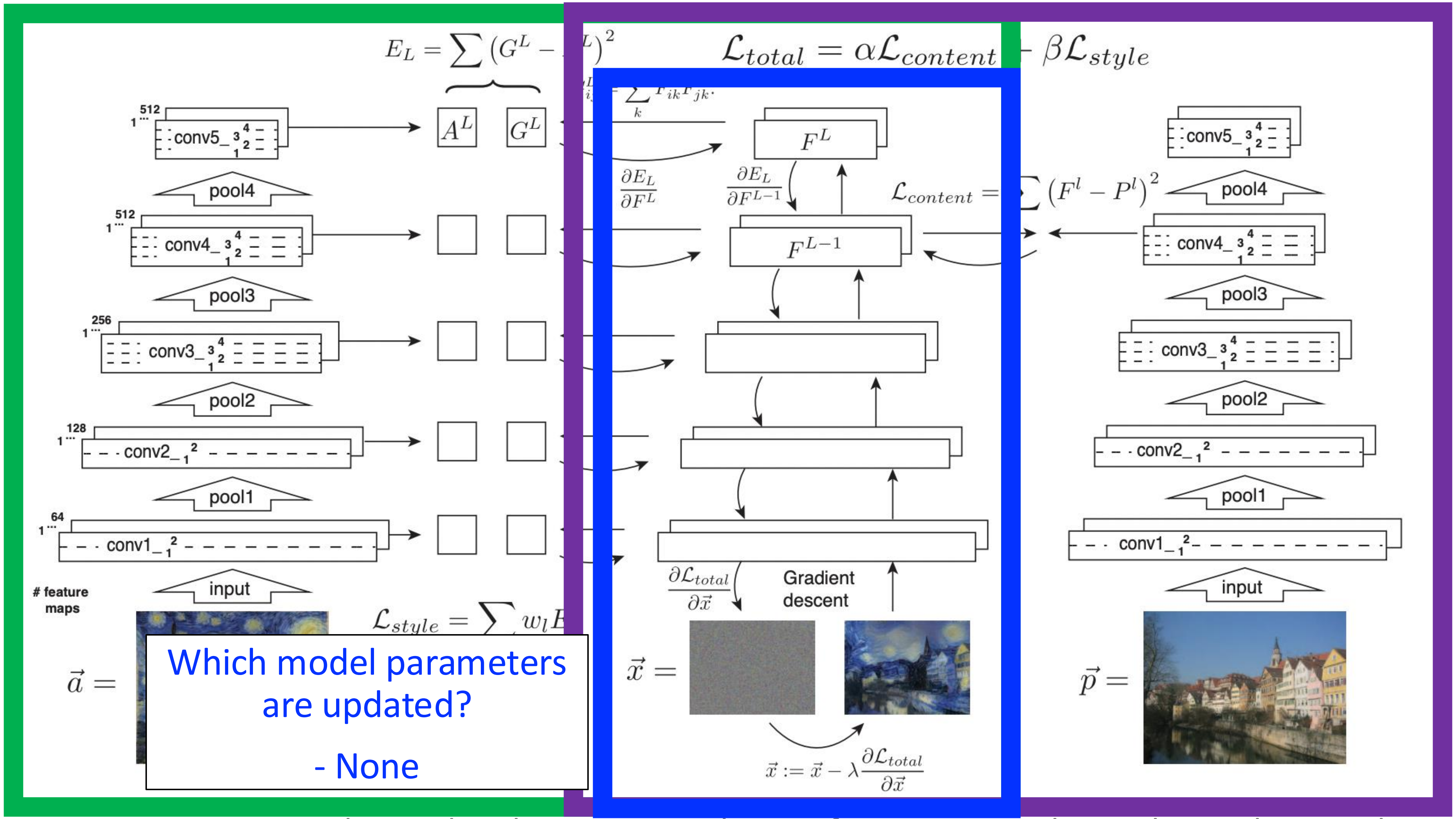

Back-propagation used to assign blame to pixels for errors and then pixels are updated

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST): Algorithm

Compute content loss based on feature maps from 1 layer

Compute style loss based on feature maps from 5 layers

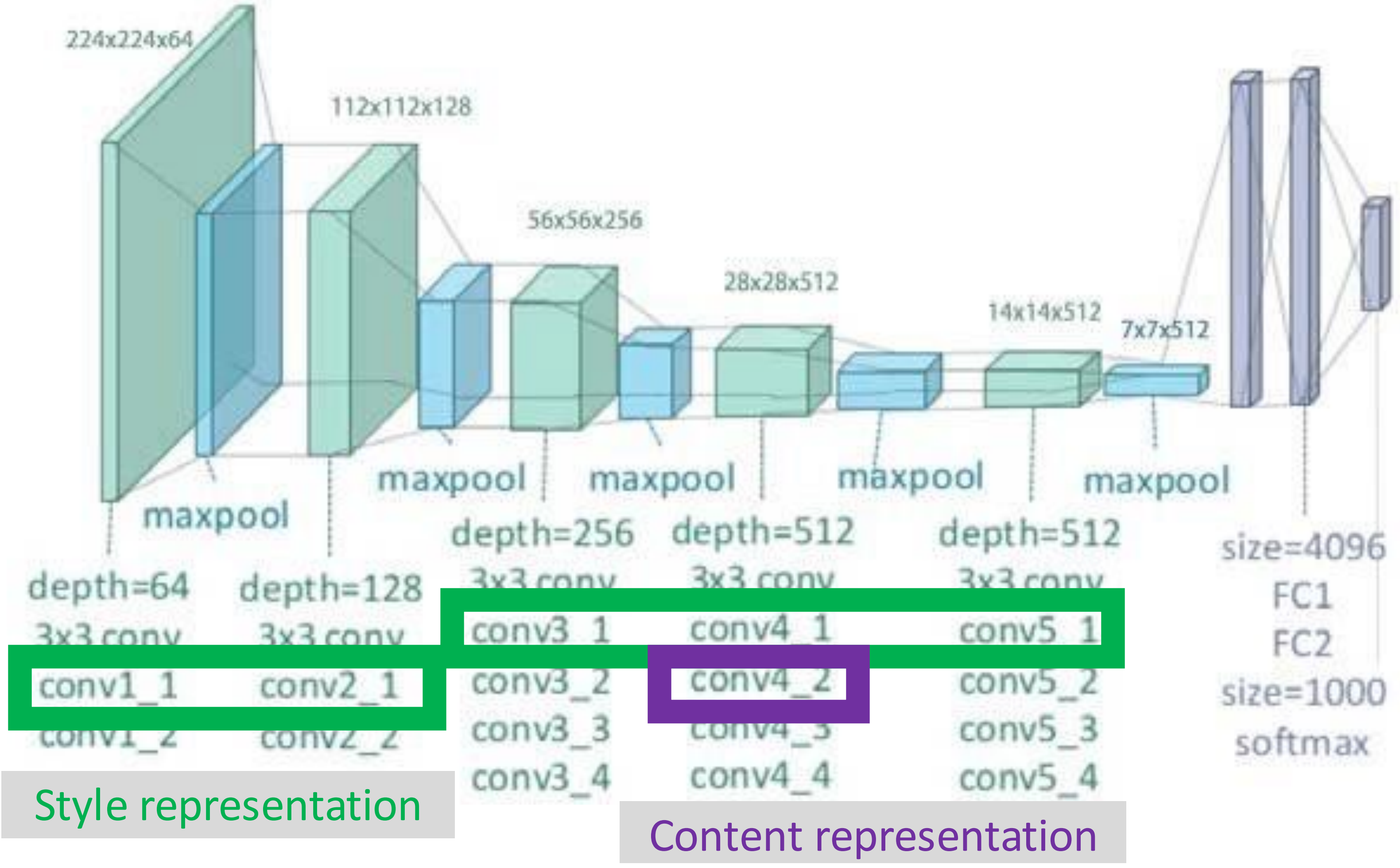



Which model parameters are updated?

- None

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST): Implementation




Uses VGG-19 for feature extraction

Figure Source: https://towardsdatascience.com/making-deep-learning-your-artist-with-style-transfer-4854055f79b7

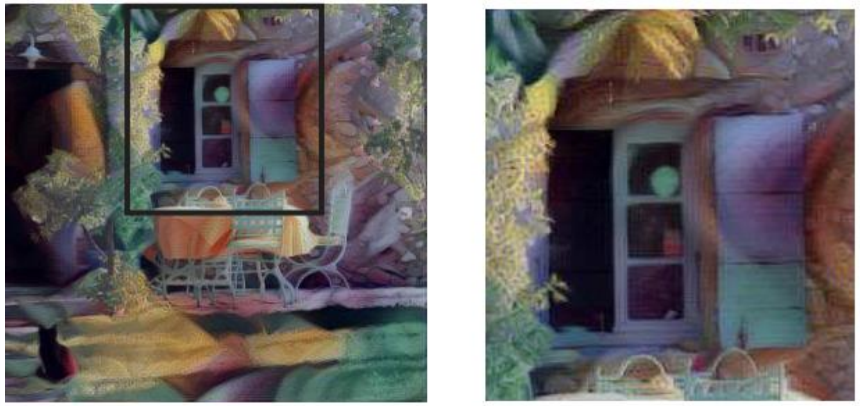
# Neural Style Transfer (NST): Influence of Different CNN Layers in Representing Content

Content image

Style image

What are the differences in the stylized results?

2nd convolutional layer of VGG-19



4th convolutional layer of VGG-19

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

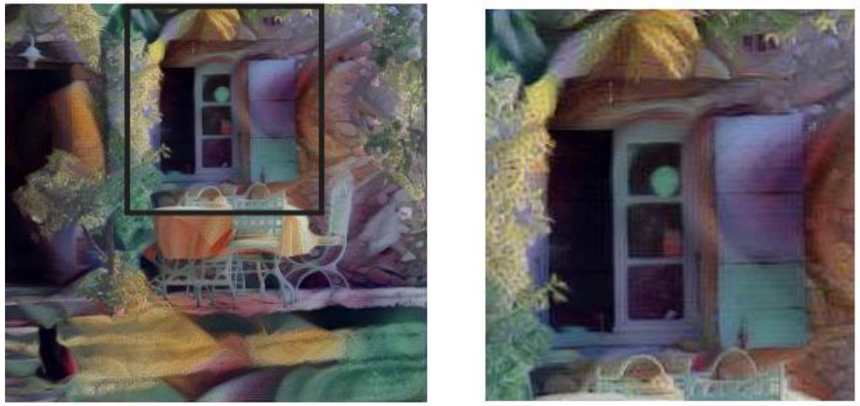
# Neural Style Transfer (NST): Influence of Different CNN Layers in Representing Content

Content image

Style image

Which result do you prefer for artistic style transfer?

$2^{nd}$ convolutional layer of VGG-19



$4^{th}$ convolutional layer of VGG-19

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

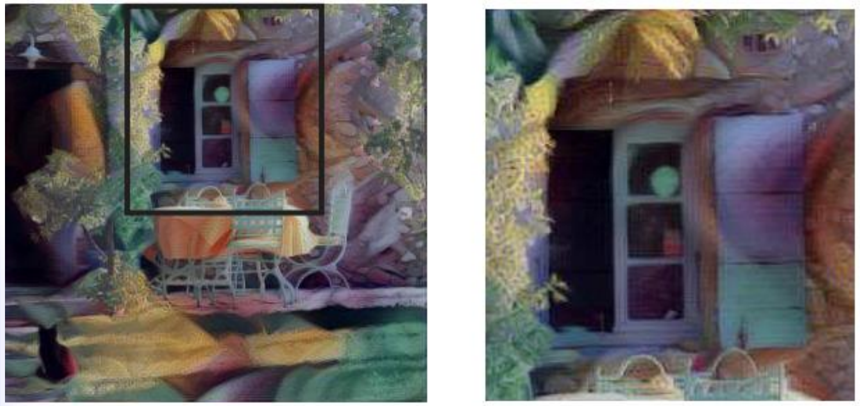
# Neural Style Transfer (NST): Influence of Different CNN Layers in Representing Content

Content image

Style image

Generally, both methods transfer color and texture information

2nd convolutional layer of VGG-19



4th convolutional layer of VGG-19

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

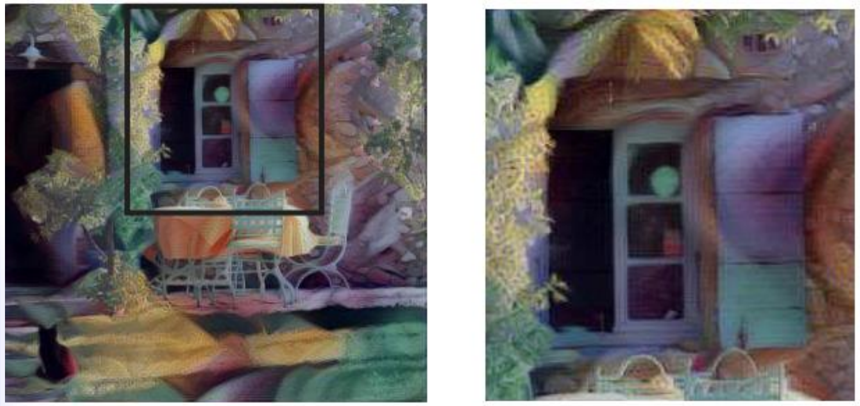
# Neural Style Transfer (NST): Influence of Different CNN Layers in Representing Content

Content image

Style image

Higher layer features lead to different colors and edges that reflect the style of the artwork without requiring rendered pixels to match those in the content image

2nd convolutional layer of VGG-19



4th convolutional layer of VGG-19

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST): Trade-off When Optimizing for Style Loss vs Content Loss

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \boxed{\alpha}\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \boxed{\beta}\mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

Greater focus on minimizing content loss

$$\alpha / \beta$$

Greater focus on minimizing style loss

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST): Trade-off When Optimizing for Style Loss vs Content Loss

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \boxed{\alpha}\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \boxed{\beta}\mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

What visual qualities arise from this style/content trade-off?

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST): Trade-off When Optimizing for Style Loss vs Content Loss

$$\mathcal{L}_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \boxed{\alpha}\mathcal{L}_{\text{content}}(\vec{p}, \vec{x}) + \boxed{\beta}\mathcal{L}_{\text{style}}(\vec{a}, \vec{x})$$

What ratio should be used to balance style and content?

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

# Neural Style Transfer (NST): Intuition Behind Findings

Can separate the content's representation because, when the CNN trains for the object recognition task, it learns to ignore image variations that can occur when recognizing an object.

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

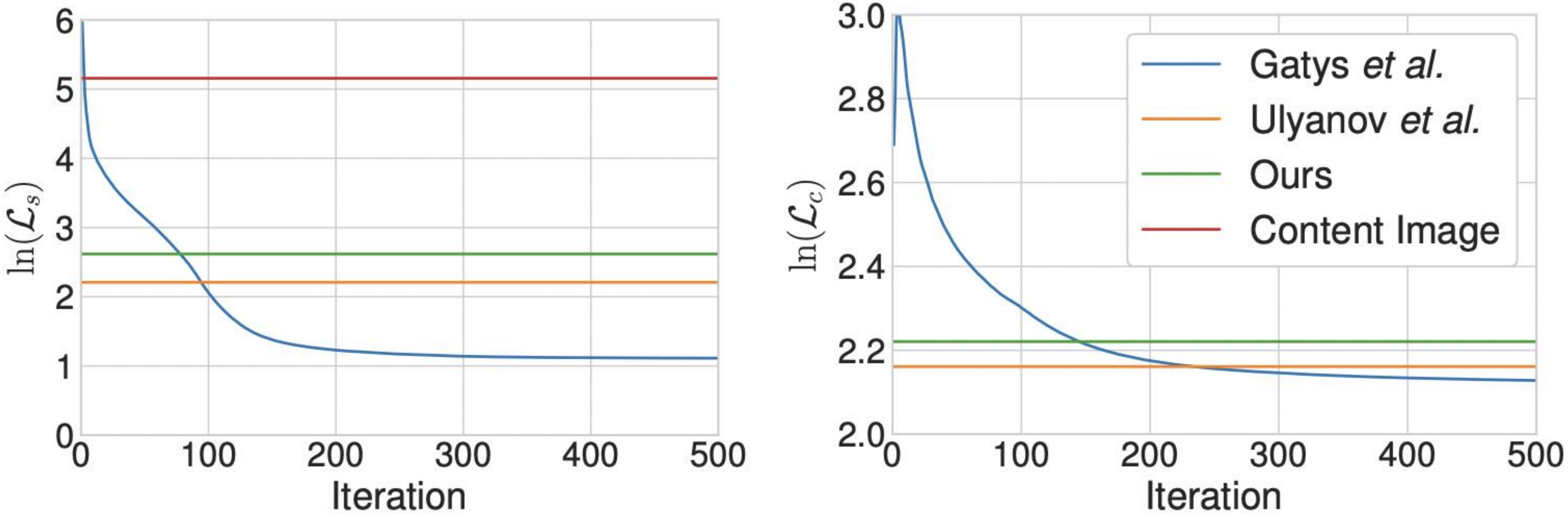
# Neural Style Transfer (NST): Intuition Behind Findings

More concisely, a representation learned for discrimination can be useful for generation

# Style Transfer: Today's Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches

# Losses Used During Training: Content and Style




(a) Style Loss

(b) Content Loss

Are higher or lower loss values better?

Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. ICCV 2017

# Common Automatic Quality Metrics

- SSIM
- FSIM
- NIMA
- BRISQUE
- NIQUE

# Human Assessment:
# "Which Carries the Style Better?"

A

B

C

D

E

Yijun Li et al. Universal Style Transfer via Feature Transforms. Neurips 2017

# Human Assessment:
# “Which is Your Favorite for a Style?”

A

B

C

D

E

Yijun Li et al. Universal Style Transfer via Feature Transforms. Neurips 2017

# Human Assessment:
# "Which Looks More Like a Real Photo?"

A

B

[Chiu and Gurari. WACV 2022]

# Human Assessment: “Which Looks More Like a Real Photo?”

A

B

[Chiu and Gurari. WACV 2022]

# Speed and Size

| Model | (a) Size | | (b) Speed performance | | | | |
|---|---|---|---|---|---|---|---|
| | # par | # layer | 1024×512 | HD 1280×720 | FHD 1920×1080 | QHD 2560×1440 | 4K 3840×2160 |
| PhNAS | 40.24M | 35 | 0.23 | OOM | OOM | OOM | OOM |
| WCT$^2$ | 10.12M | **24** | 0.30 | 0.43 | 0.80 | OOM | OOM |
| PhWCT | 8.35M | 48 | 0.21+0.03 | 0.32+0.06 | 0.61+0.14 | 1.01+0.23 | OOM |
| Ours (E2E) Ours (BT) | **7.05M** | **24** | **0.18+0.03** | **0.24+0.06** | **0.39+0.14** | **0.59+0.23** | **1.22+0.54** |

Want model to run faster across many resolutions
(and so typically have fewer parameters)

[Chiu and Gurari. WACV 2022]

# Style Transfer: Today's Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches

# Neural Style Transfer (NST): Limitation

Slow; for example, synthesizing a 512x512 image takes ~1 hour

(it requires *iterative* optimization)

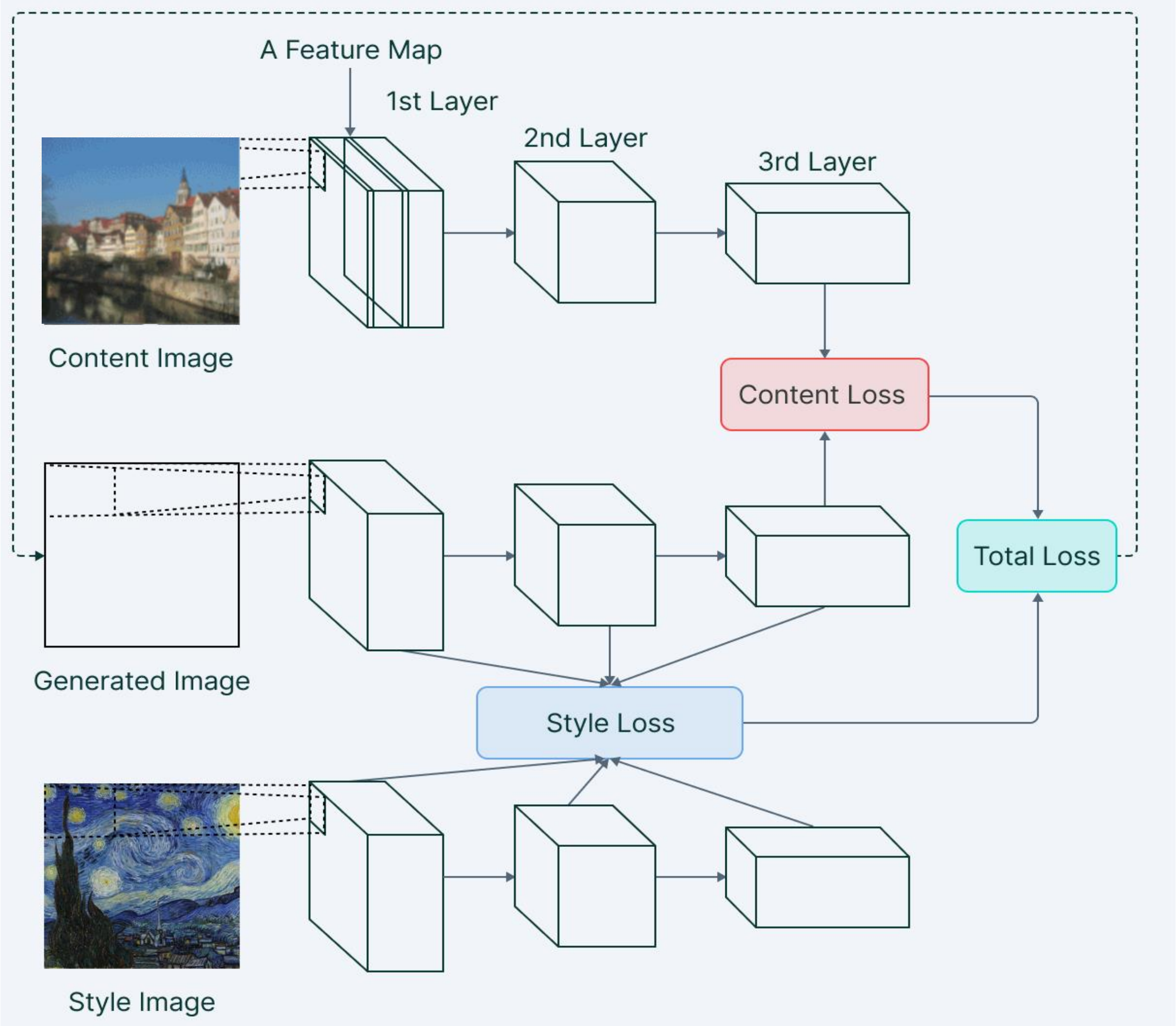


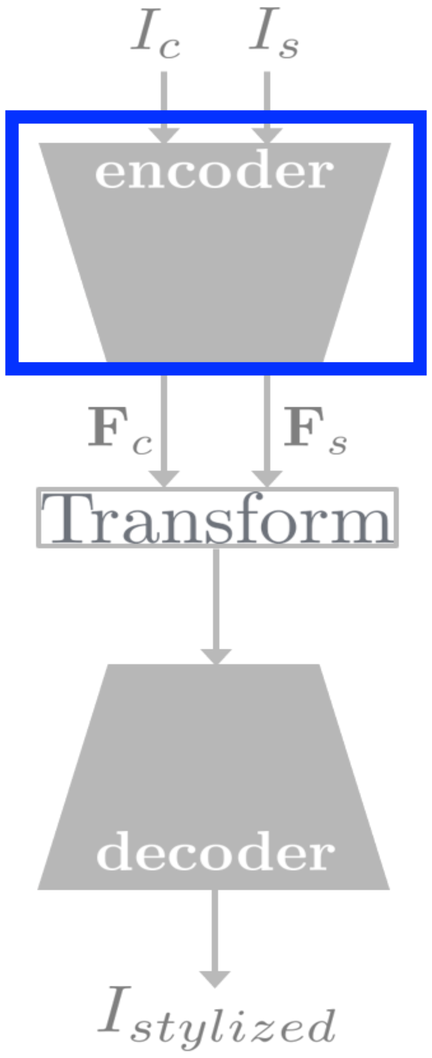

Figure Sources: https://ndres.me/images/style-transfer.gif; https://www.v7labs.com/blog/neural-style-transfer

# Autoencoders

- How to computationally isolate the content of the content image?
- How to computationally isolate the style of the style image?
- How to blend the content and style?
- How to support any arbitrary style?
- How to do all these fast?

Gatys, Ecker, and Bethge. Image Style Transfer Using Convolutional Neural Networks. CVPR 2016

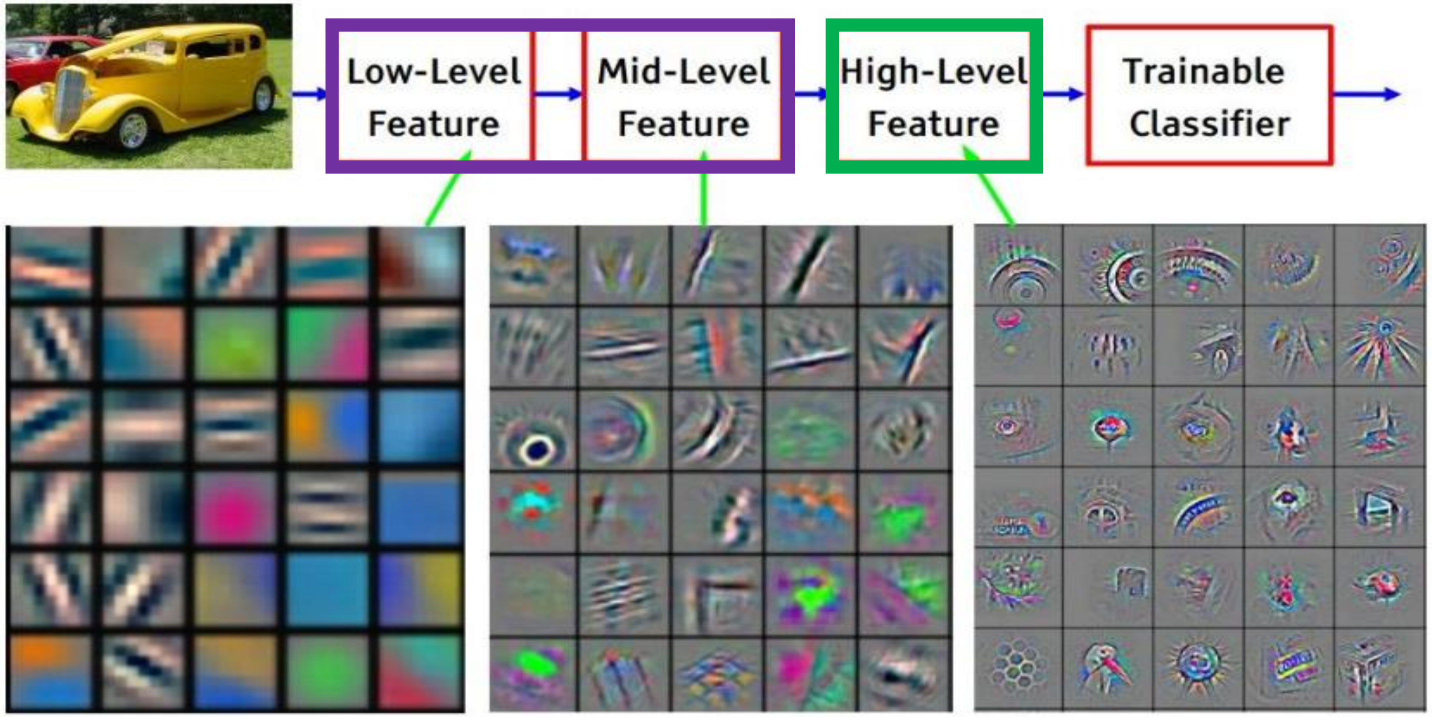
# Background: Autoencoder Solution




Extracts content and style features; e.g., VGG-19

**Style**: statistics summarize features in multiple layers

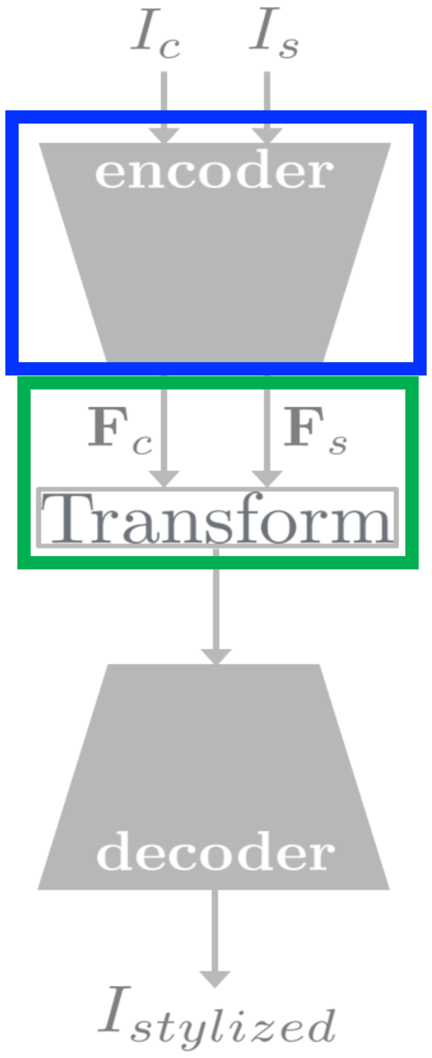
**Content**




Source: [Chiu and Gurari. ECCV 2020]

Source: Yann LeCun

# Background: Autoencoder Solution



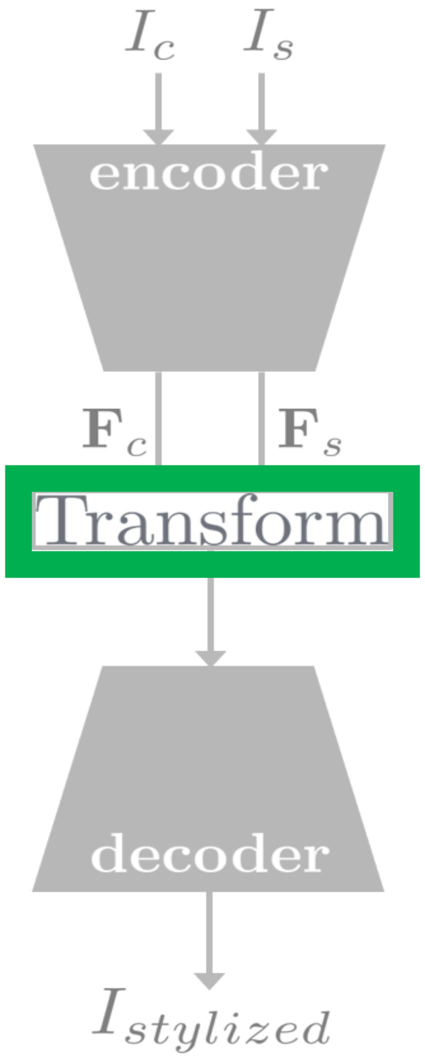

Extracts content and style features

Content features transformed to match statistics of style features (no training needed)

Resulting feature decoded into a stylized image

Source: [Chiu and Gurari. ECCV 2020]

# Autoencoder: Transformation Types




Global first- and second-order transformations:

e.g., WCT adjusts covariance

e.g., AdaIn adjusts mean and variance

Global higher-order statistics…

e.g., Kalischek et al. CVPR 2021

e.g., Zhang et al. CVPR 2022

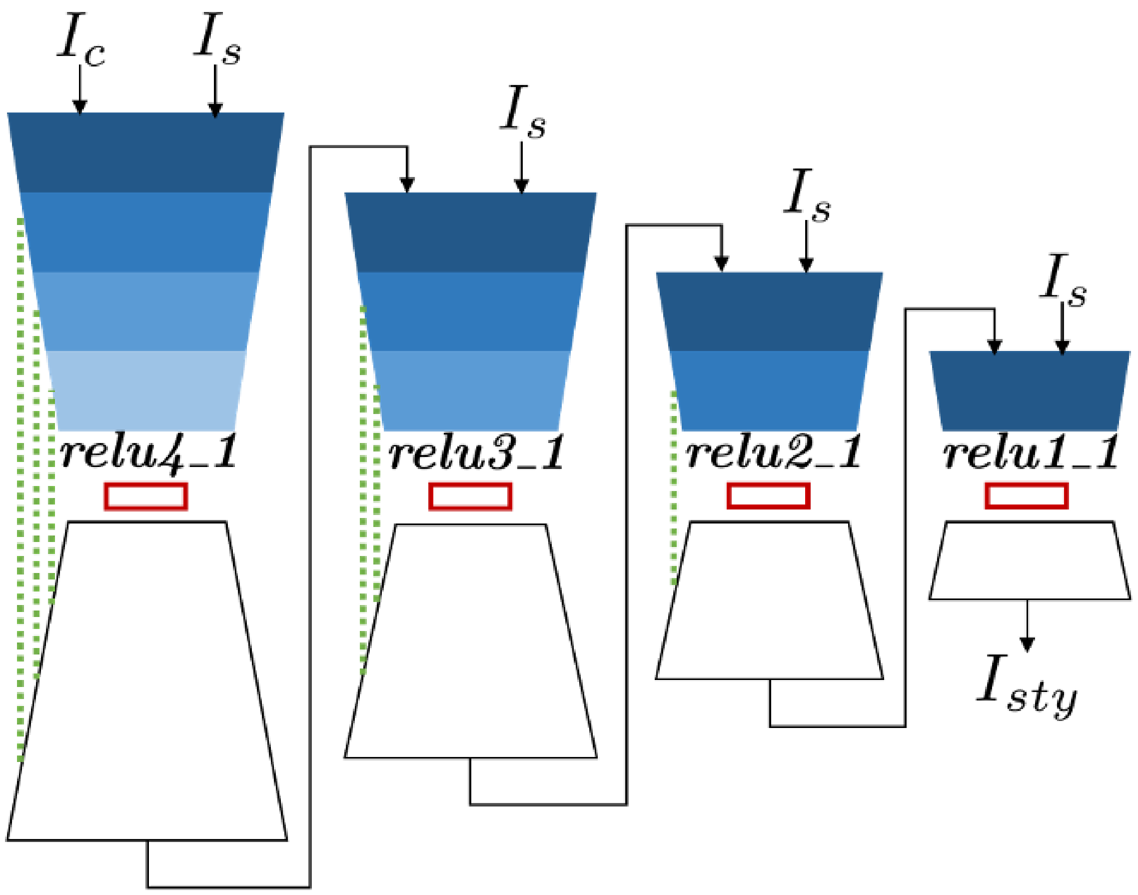
Local statistics

e.g., StyleSwap adjusts patches


Yijun Li et al. Universal Style Transfer via Feature Transforms. Neurips 2017

Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. ICCV 2017

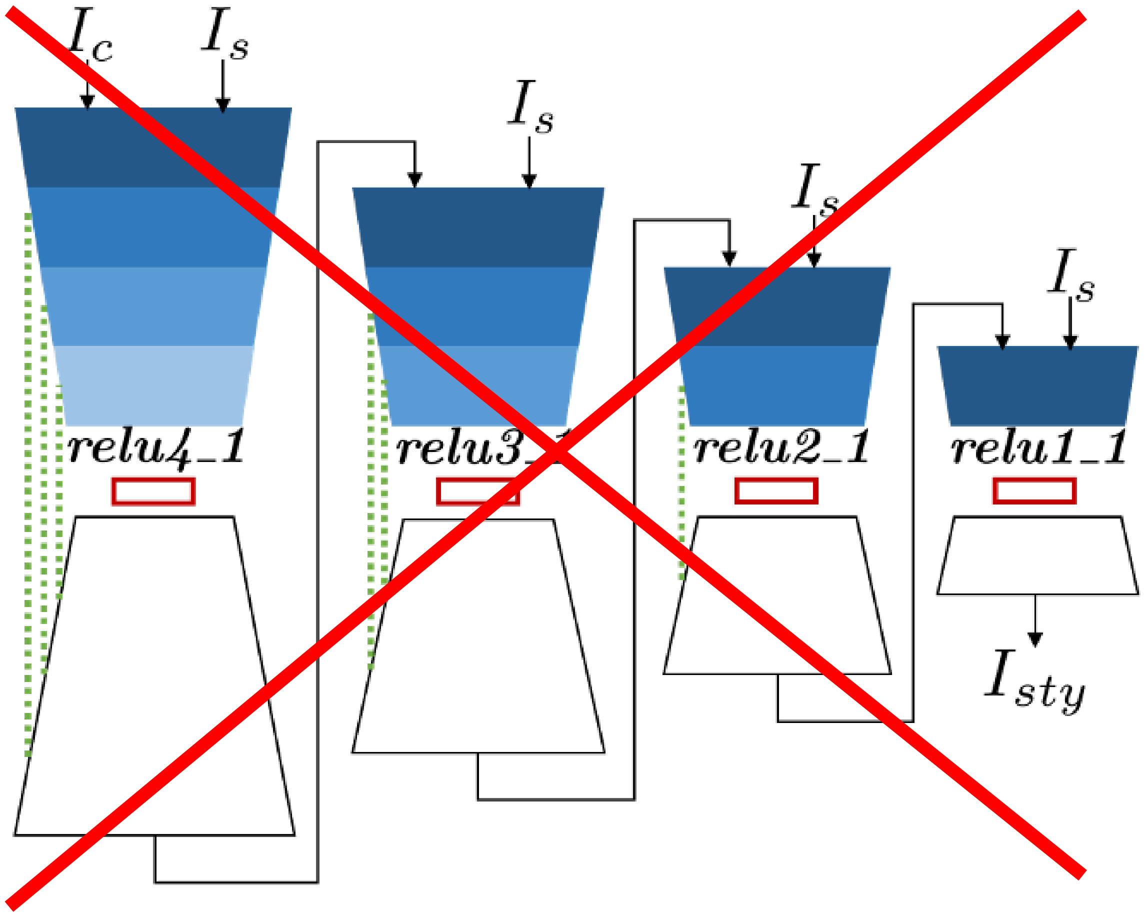
# Autoencoder: Transform Multiple Layers to Achieve Stronger Stylization for Coarse and Fine Features

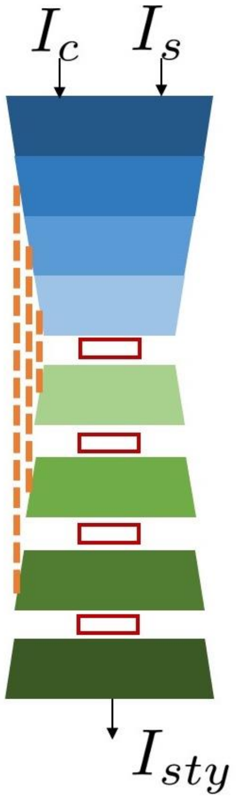



Cascade of autoencoders gradually fine-tune content image with respect to the style image for coarse to fine features

[Chiu and Gurari. WACV 2022]

# Autoencoder: Transform Multiple Layers to Achieve Stronger Stylization for Coarse and Fine Features




(decoders produce target encoder features for coarse-to-fine feature transformation)

Compact model can achieve comparable results with 30.3% fewer parameters while supporting higher resolution images (4K) and achieving faster stylization!

[Chiu and Gurari. WACV 2022]

# Style Transfer: Today’s Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches

# Other Networks Address…

- How to computationally isolate the content of the content image?
- How to computationally isolate the style of the style image?
- How to blend the content and style?
- How to support any arbitrary style?
- How to do all these fast?

# e.g., Prompting With Trained Style Word Vectors

Prompt: "a $S_i$ style of a [class]"

Cho et al. PromptStyler: Prompt-driven Style Generation for Source-free Domain Generalization. ICCV 2023

# Style Transfer: Today’s Topics

- Problem
- Applications
- Neural Style Transfer Model
- Evaluation Metrics
- Autoencoder-Based Models
- Other Approaches


The End