PyTorch Seminar

Day4. Neural Style Transfer

Duhyeon Kim / May 2025

PyTorch

PyTorch Seminar

Day4. Neural Style Transform

Duhyeon Kim / May 2025

PyTorch

Contents

Model = Artists

A. Paper Introduction

B. Feature map (Vgg-19)

C. Training Method

D. Two different loss function

E. Results

Image Style Transfer Using Convolutional Neural Networks

Gatys et al. CVPR 2016

Image Style Transfer Using Convolutional Neural Networks

Leon A. Gatys
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Graduate School of Neural Information Processing, University of Tübingen, Germany
leon.gatys@bethgelab.org

Alexander S. Ecker
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany
Baylor College of Medicine, Houston, TX, USA

Matthias Bethge
Centre for Integrative Neuroscience, University of Tübingen, Germany
Bernstein Center for Computational Neuroscience, Tübingen, Germany
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Paper Introduction

Gatys et al. CVPR 2016

TITLE	CITED BY	YEAR
Image style transfer using convolutional neural networks LA Gatys, AS Ecker, M Bethge Proceedings of the IEEE conference on computer vision and pattern …	10406 *	2016

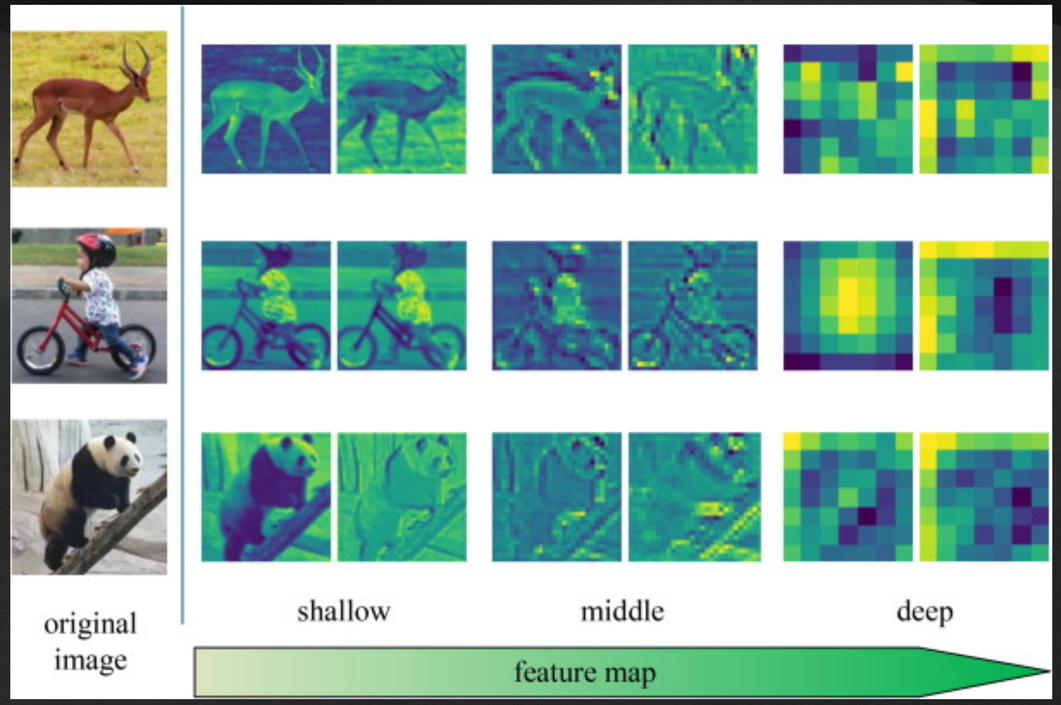
Leon A. Gatys

Apple Inc.
Verified email at bethgelab.org

Machine Learning for Health Deep Le
Style Transfer

Content Loss
+ Style Loss

Fixed Model Parameter



Recap

Key difference from formal training

Formal

$$\theta \leftarrow \theta - \alpha \frac{\partial L}{\partial \theta}$$

Parameter update

Neural Style Transfer

$$x \leftarrow x - \alpha \frac{\partial L}{\partial x}$$

Pixel Value update

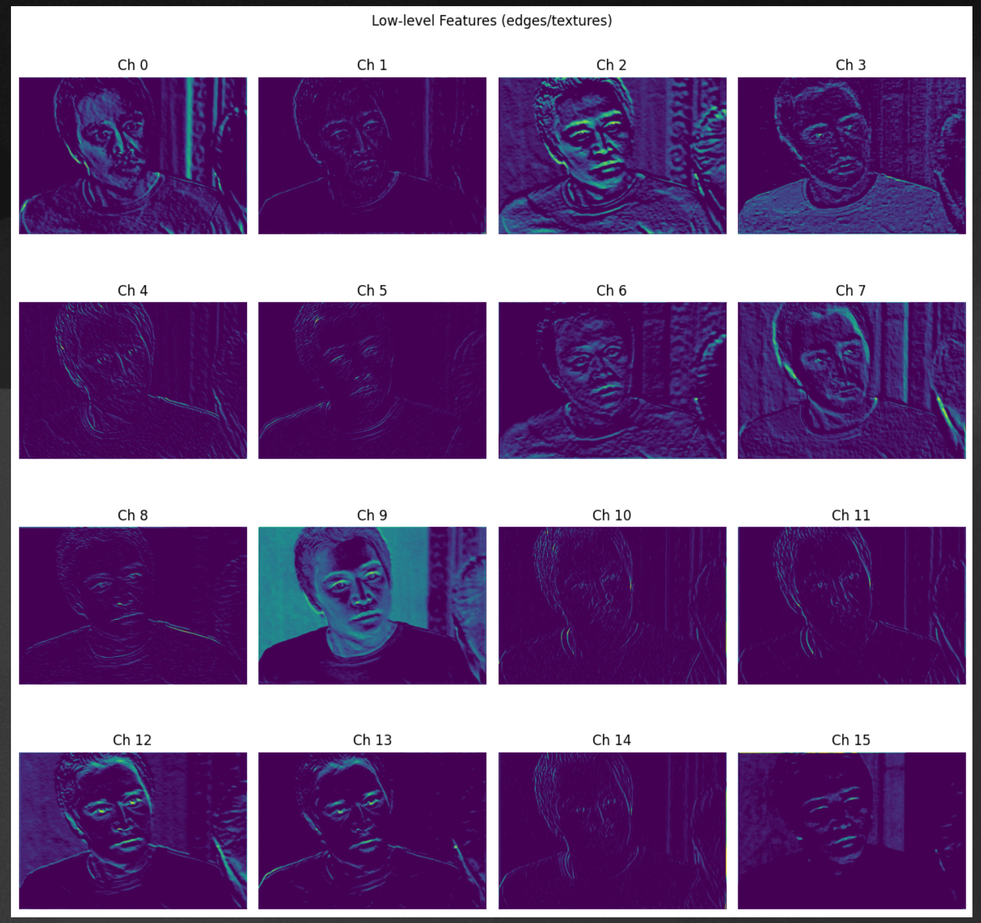
Vgg19 network

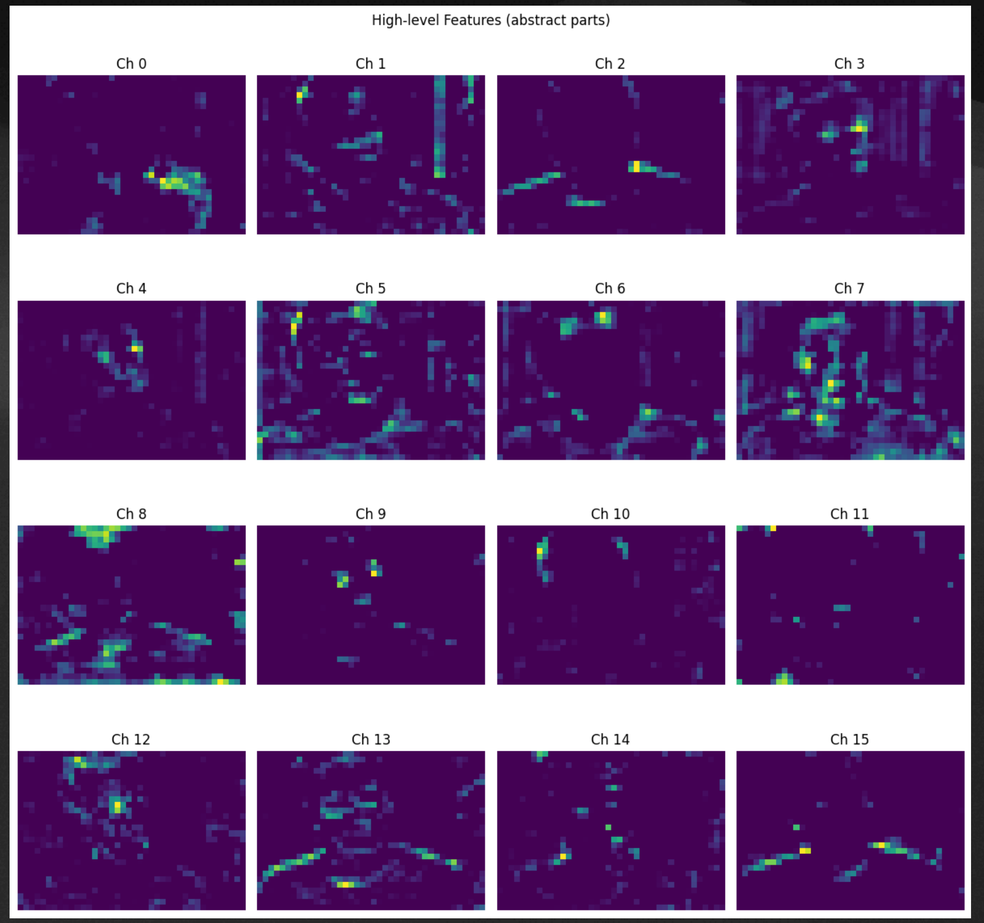
Classifier (1000 classes)

```
lakeside: 0.3555
pier: 0.1757
breakwater: 0.1575
steel arch bridge: 0.0619
seashore: 0.0494
VGG19 output shape: torch.Size([1, 1000])
```

CNN feature map

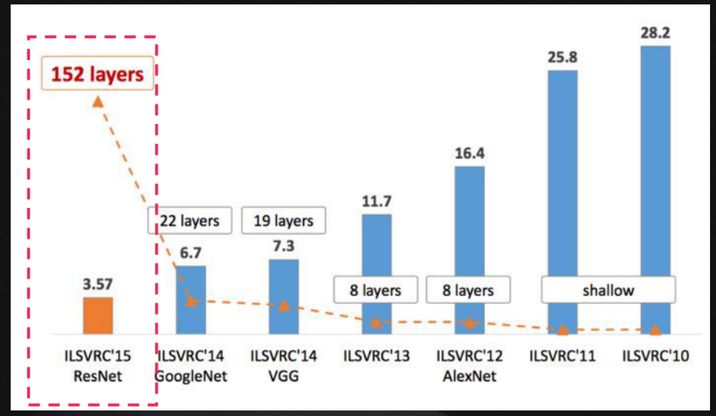
Visualization



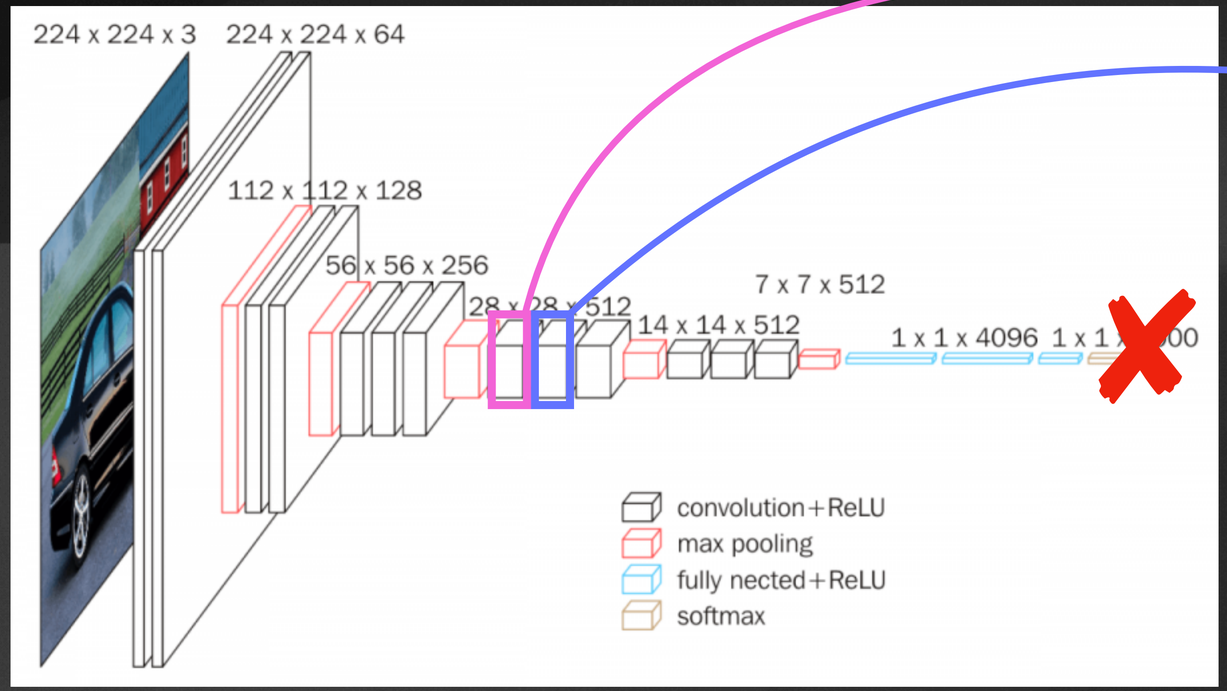


Vgg-19 Network

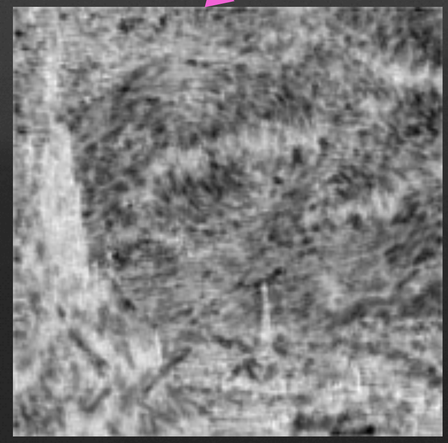
Feature map



Robust perceptual feature extractor



Content



Style (some magic)

Vgg-19 Network

Inference input form (when using PRETRAINED model)

The inference transforms are available at `VGG19_Weights.IMAGENET1K_V1.transforms` and perform the following preprocessing operations: Accepts `PIL.Image`, batched `(B, C, H, W)` and single `(C, H, W)` image `torch.Tensor` objects. The images are resized to `resize_size=[256]` using `interpolation=InterpolationMode.BILINEAR`, followed by a central crop of `crop_size=[224]`. Finally the values are first rescaled to `[0.0, 1.0]` and then normalized using `mean=[0.485, 0.456, 0.406]` and `std=[0.229, 0.224, 0.225]`.

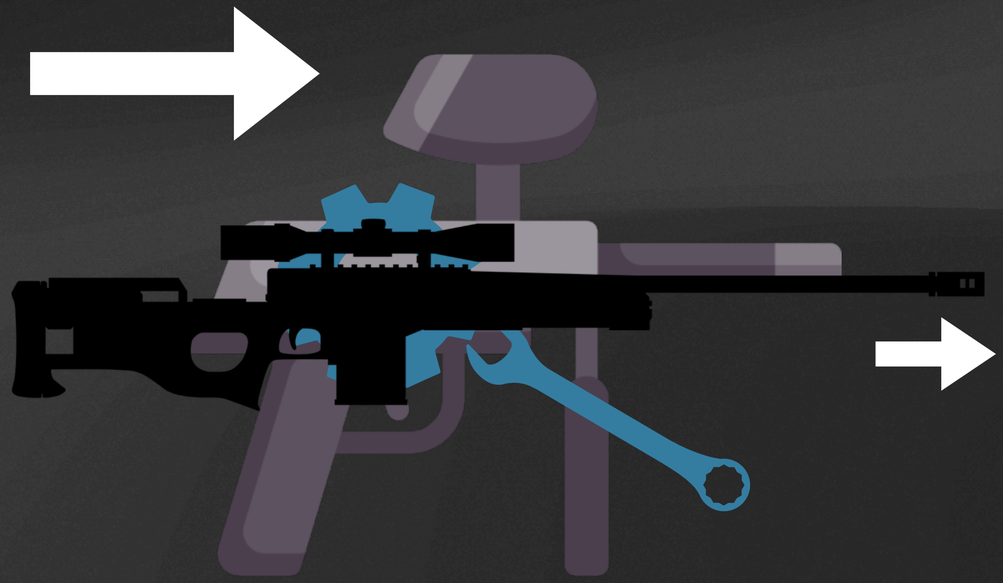
Understanding the data format that a model was trained on (such as channel order like RGB vs BGR, and normalization schemes) is crucial when utilizing the model.

https://docs.pytorch.org/vision/main/models/generated/torchvision.models.vgg19.html

Training Methodology

Conventional training

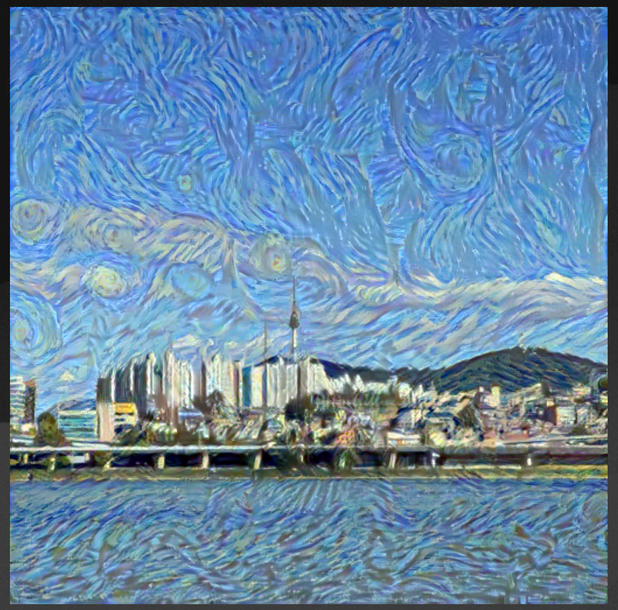
$$\theta \leftarrow \theta - \alpha \frac{\partial L}{\partial \theta}$$

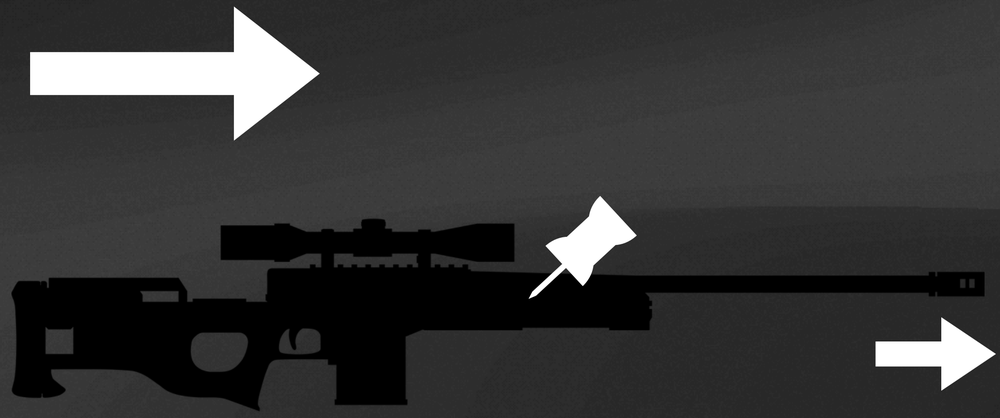


```
lakeside: 0.3555
pier: 0.1757
breakwater: 0.1575
steel arch bridge: 0.0619
seashore: 0.0494
VGG19 output shape: torch.Size([1, 1000])
```

Training Methodology

Neural Style Transfer





Vgg19 net

```
lakeside: 0.3555
pier: 0.1757
breakwater: 0.1575
steel arch bridge: 0.0619
seashore: 0.0494
VGG19 output shape: torch.Size([1, 1000])
```

$$x \leftarrow x - \alpha \frac{\partial L}{\partial x}$$

Difference is optimization method & …

```
optimizer = torch.optim.LBFGS([generated_image])
optimizer = torch.optim.SGD(classifer.parameters(), lr=0.01, momentum=0)
```

Loss functions

Content and Style loss

Point 1. We nee to synthesize the style(texture) of the style image.

- min diff of. [Style image style ↔️ Transferred image style]

Point 2. We need to preserve content image's original content(structure).

- min diff of. [Original image content ↔️ Transferred image content]

Style image style ↔️ Transferred image style

Point 1. Style Loss

What is "style"?
How to quantify

(d,e). **Style Reconstructions**. On top of the original CNN activations we use a feature space that captures the texture information of an input image. The style representation computes correlations between the different features in different layers of the CNN. We reconstruct the style of the input image from a style representation built on different subsets of CNN layers ('conv1_1' (a), 'conv1_1' and 'conv2_1' (b), 'conv1_1', 'conv2_1' and 'conv3_1' (c), 'conv1_1', 'conv2_1', 'conv3_1' and 'conv4_1' (d), 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' (e). This creates images that match the style of a given image on an increasing scale while discarding information of the global arrangement of the scene.

Minimize the distance of style representation layers btw style image and transferred image.

Style image style ↔️ Transferred image style

Point 1. Style Loss

Minimize the distance of style representation layers btw style image and transferred image.

Structure ⊄ Style

Layer Output (b,c,h,w) − Structure = Gram Matrix

$$G^{(b)} = F^{(b)}(F^{(b)})^T, \quad \text{where} \quad F^{(b)} \in \mathbb{R}^{C\times(HW)}, \quad b = 1,\ldots,B$$

Minimize the distance of Gram Matrix (style layer's) btw style image and transferred image.

Style image style ↔️ Transferred image style

Point 1. Style Loss

Minimize the distance of Gram Matrix (style layer's) btw style image and transferred image.

Distance = L2 = MSE

$$L_{\text{style}}(\mathbf{G}, \mathbf{S}) = \frac{1}{N} \sum_{l=1}^{N} \left\| G^{(l)}(\mathbf{G}) - G^{(l)}(\mathbf{S}) \right\|_F^2$$

Original image content ↔ Transferred image content

Point 2. Content Loss

What is
How to quantify “content”?

Content = Structures

Structures = edges + objects + …

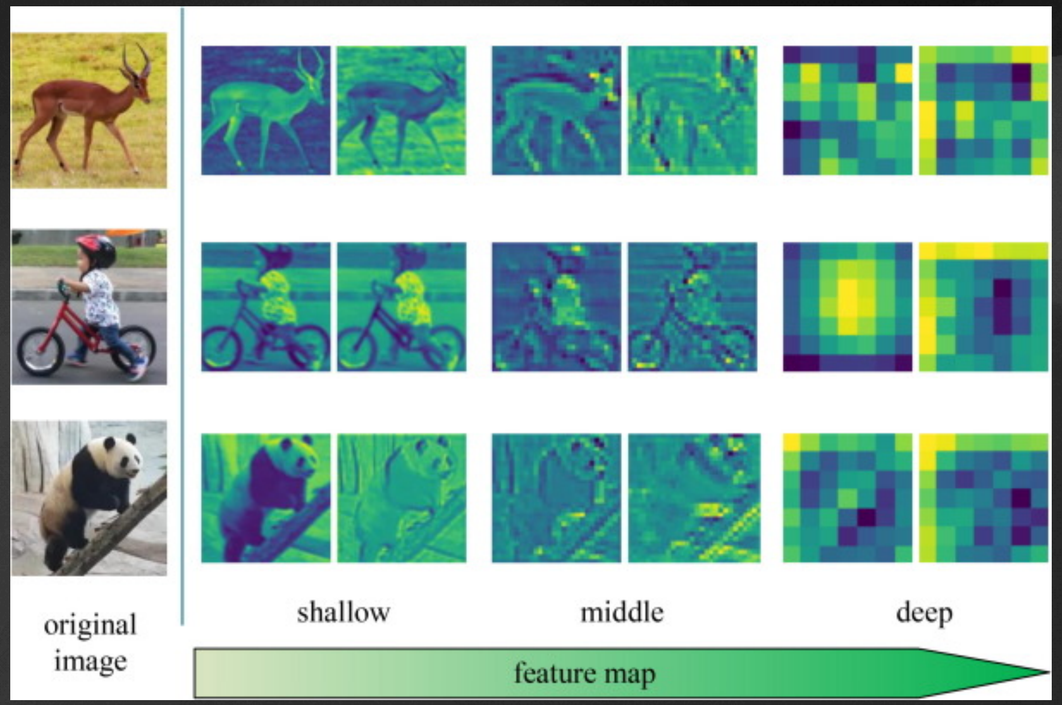
Feature maps’ spatial information

Content can be quantified by Feature maps’ spatial information

Original image content ↔️ Transferred image content

Point 2. Content Loss

Content can be quantified by Feature maps' spatial information

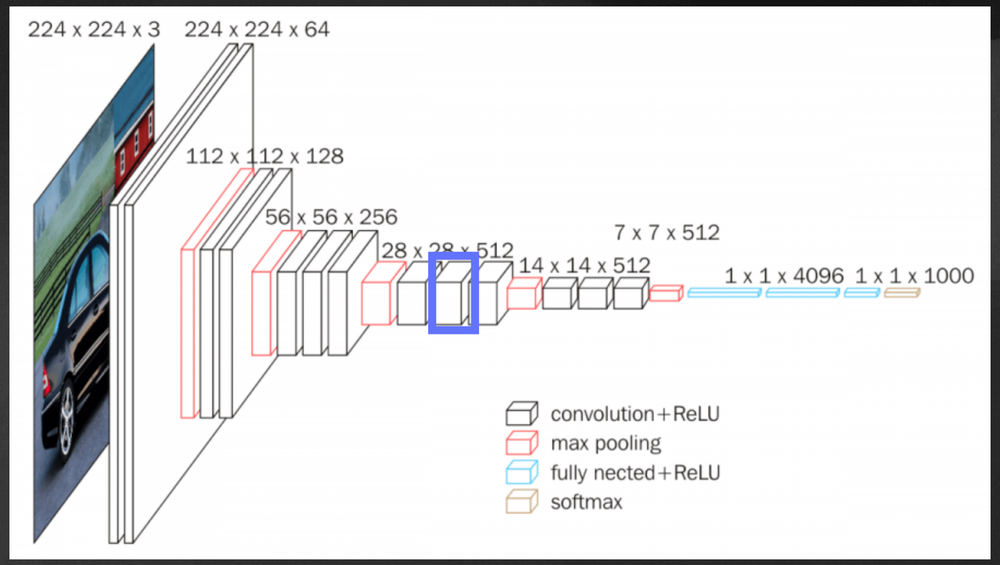


By experiment, conv4_2 is selected to be structure information

Original image content ↔️ Transferred image content

Point 2. Content Loss

By experiment, conv4_2 is selected to be structure information



$$L_{\text{content}}(G, C) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^{\text{conv4\_2}}(G) - F_{ij}^{\text{conv4\_2}}(C) \right)^2$$

Total Loss

Style loss + Content loss

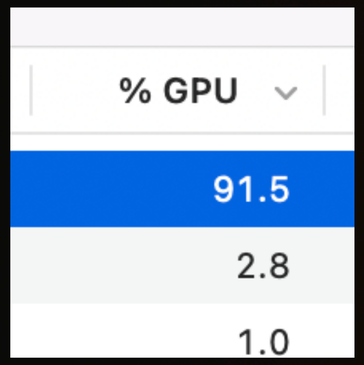
$$L_{\text{style}}(\mathbf{S}, \mathbf{G}) = \frac{1}{N} \sum_{l=1}^{N} \left\| G^{(l)}(\mathbf{G}) - G^{(l)}(\mathbf{S}) \right\|_F^2$$

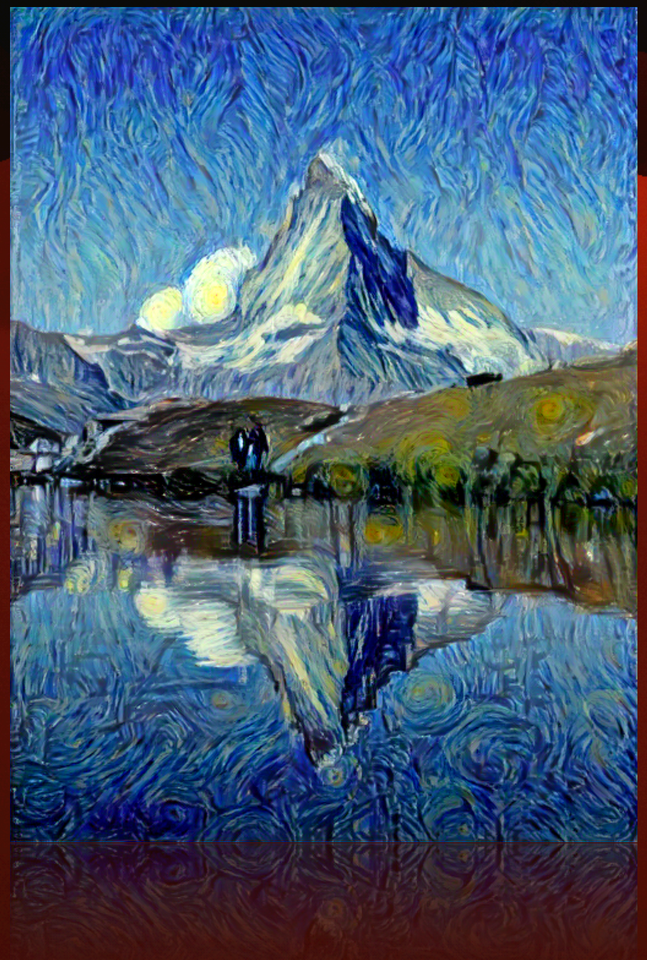
$$L_{\text{content}}(C, G) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^{\text{conv4\_2}}(G) - F_{ij}^{\text{conv4\_2}}(C) \right)^2$$

$$L_{\text{total}}(\vec{p}, \vec{a}, \vec{x}) = \alpha \cdot L_{\text{content}}(\vec{p}, \vec{x}) + \beta \cdot L_{\text{style}}(\vec{a}, \vec{x})$$

With this loss function we update generated image. $x \leftarrow x - \alpha \frac{\partial L}{\partial x}$

Neural Style Transfer in Code





Assignment #3

Your own Style Transfer Image

Apply Neural Style Transfer (NST) to a photo you took yourself.
Transform it using any style image of your choice.

🧾 Submission Requirements

- 📁 Jupyter Notebook (.ipynb)
- 🖼️ Content Image (Original)
- 🎨 Style Image
- 🖌️ Stylized Output Image

Thank you

Summary

Overall Code Structure

CNN

MLP

1 2 3 4 5 6 7 8 9 0

Predicted

```
{
    'cab': 0.08239,
    'dog': 0.04366,
    'cat': 0.09582,
    'car': 0.22994,
    'bird': 0.03967,
    'plane': 0.03967,
    'boat': 0.24323,
    'tree': 0.10801,
    'horse': 0.03135,
    'bike': 0.08626
}
```

Answer

```
{
    'cab': 1,
    'dog': 0,
    'cat': 0,
    'car': 0,
    'bird': 0,
    'plane': 0,
    'boat': 0,
    'tree': 0,
    'horse': 0,
    'bike': 0
}
```

Dataloader

CustomDataset

batches

Cross Entropy Loss

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

$$y_j = 1, \quad y_{i \neq j} = 0$$

$$L = -\log(\hat{y}_j)$$

SGD optimizer

$$\theta_{new} = \theta_{old} - \eta \cdot \nabla_\theta \cdot lr$$