

Arbitrary Style Transfer

Computer Vision

D'ORAZIO ANTONIO, 1967788

MINUT ROBERT ADRIAN, 1942806

ZARBA MELI GIACOMO, 1807439





Overview

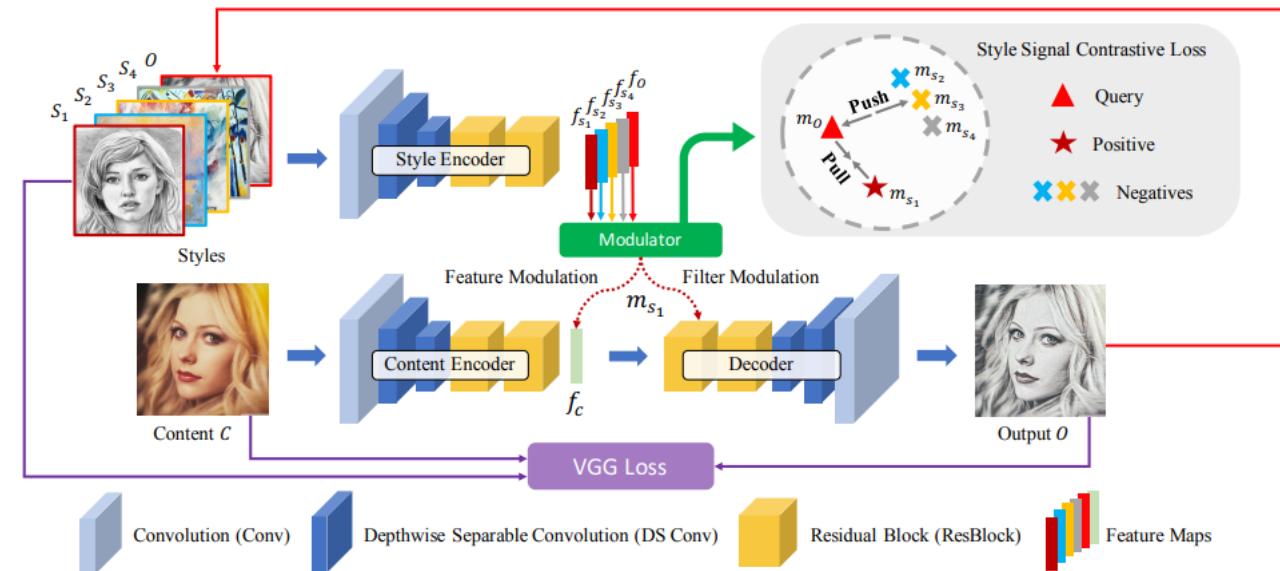
- **Code refactoring with newer libraries**
PyTorch Lightning
Torchmetrics
Weights and Biases
- **Different experiments** to improve the performance from the baseline code
Architecture
Loss functions
- **Model fine-tuning** for two different tasks
Generic
Faces
- **Demo app** for testing the model

Reference Paper

MicroAST

Key points:

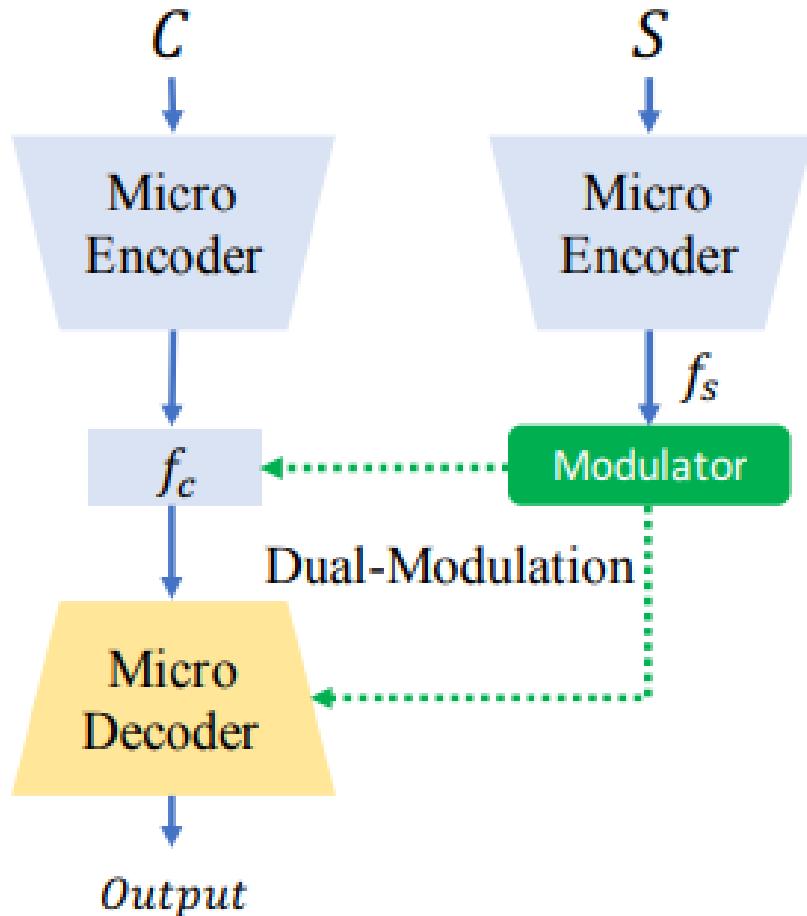
- Encode and decode content and style through **lightweight components**
Two **micro-encoders**, for encoding the content and style features
A **micro-decoder**, for applying the style on the content image
A **dual-modulator** which helps to encode and decode styles
- Use the VGG only at training time to compute the loss
- At inference time, only use the learned components



Style Signal Contrastive Loss
▲ Query
★ Positive
✖✖✖ Negatives

Inference pipeline

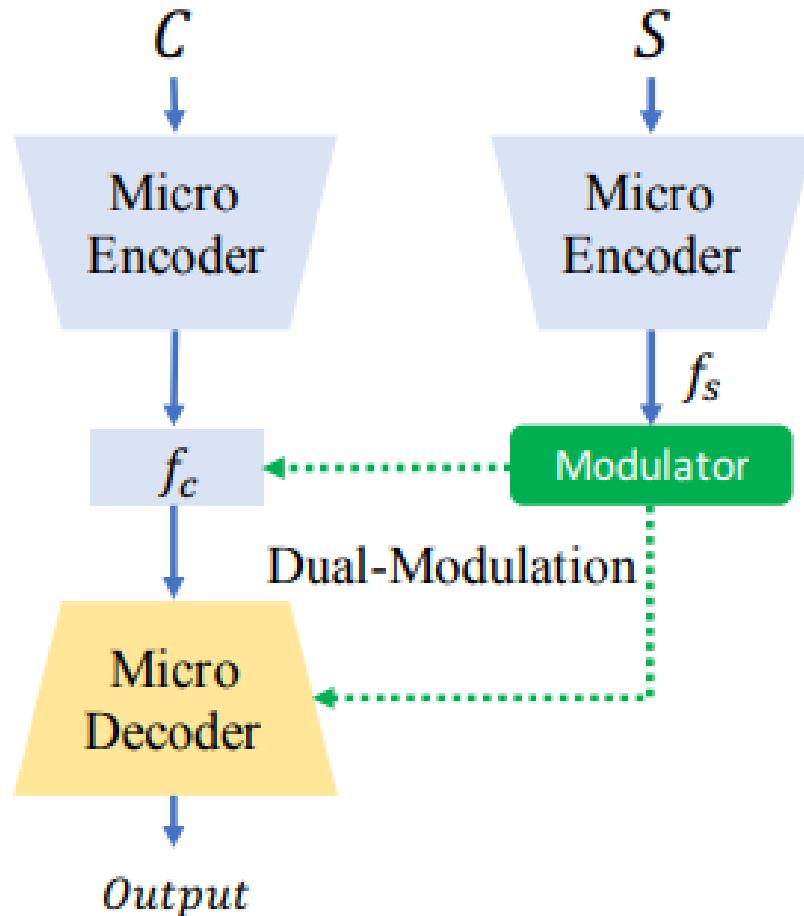
- Encode the content image
Content Encoder: $f_c := E_c(C)$
- Encode the style image
Style Encoder: $f_s := E_s(S)$
Modulator: $m_s := M_s(f_s)$
- Apply the style and generate the output
Decoder: $O_s := D(f_c, m_s)$



Encoders and decoder

Designed to be lightweight

- **Content Encoder** f_c , **Style Encoder** f_s
 - One standard stride-1 convolutional (Conv) layer
 - Two stride-2 depthwise separable convolutional (DS Conv) layers
 - Two stride-1 residual blocks (ResBlocks)
- **Decoder** D
 - Symmetrical to the encoders



The Dual-Modulator

Captures the styles signal of the encoded features

- **DualMod:**

$$DualMod(D, f_c, m_s) := FeatMod(f_c, (\mu_s, \sigma_s)) + FilterMod(D, (w_s, b_s))$$

- **FeatMod:** $FeatMod(f_c, (\mu_s, \sigma_s)) := \sigma_s \left(\frac{f_c - \mu(f_c)}{\sigma(f_c)} \right) + \mu_s$

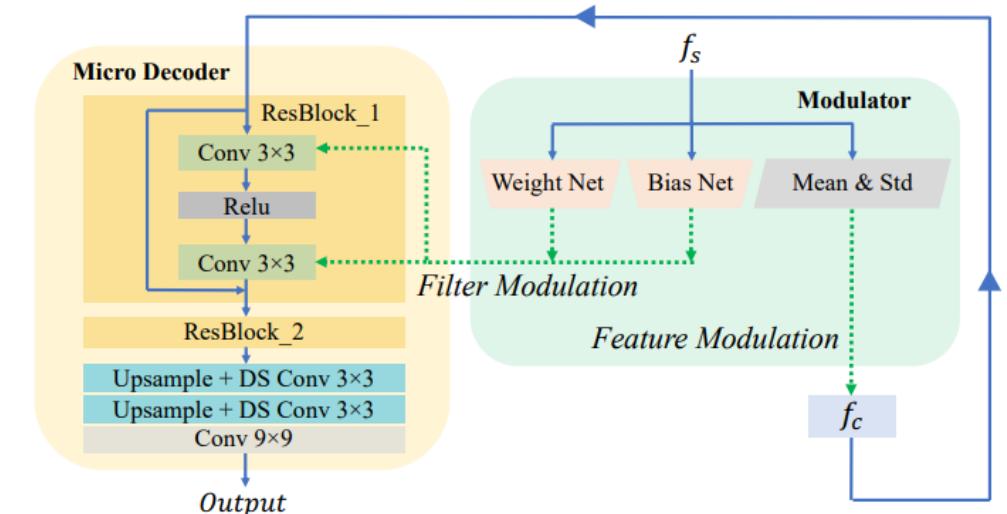
Captures the global attributes

Uses the learned channel-wise mean and std of the style features to modulate the content features

- **FilterMod:** $FilterMod(D, (w_s, b_s)) := ResBlock(f_c, (w_s, b_s))$

Captures the local textures (brushstrokes)

The result is injected into the decoder D to modulate its Conv filters in the ResBlock



Where:

$$\mu_s := \mu(f_s),$$

$$\sigma_s := \sigma(f_s),$$

$$\xi = NeuralNetwork,$$

$$w_s := \xi_w(f_s),$$

$$b_s := \xi_b(f_s),$$

$$m_s := (\mu_s, \sigma_s, w_s, b_s)$$

Paper's proposed loss

- **Content Loss** (i.e., perceptual loss)
MSE of features at layer relu4_1 of the VGG-19
- **Style Loss**
To match the Instance Normalization statistics, computed at layers {relu1_1, relu2_1, relu3_1, relu4_1}
- **Style Signal Contrastive Loss**

$$L_{contrastive} = \sum_{i=1}^N \frac{||m_{o_i} - m_{s_i}||_2}{\sum_{j \neq i}^N ||m_{o_i} - m_{s_j}||_2}$$

- **The final loss is**

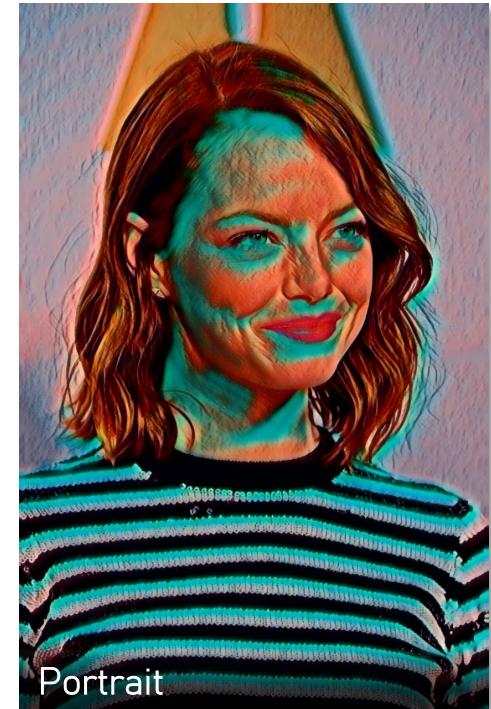
$$L_{full} := \lambda_{content} L_{content} + \lambda_{style} L_{style} + \lambda_{contrastive} L_{contrastive}$$

The model



Challenges

- MicroAST produces artifacts by upsampling with bilinear interpolation
- The style may not be applied correctly
- Doesn't work well for portraits



Experiments

- **Decoder**
Transposed Convolution
15x15 Convolution after bicubic upsampling
- **Loss**
L1
LPIPS (Learned Perceptual Image Patch Similarity)
Cosine distance
Regularization (Total Variation, Frechet distance)
- **Datasets**
LAION
FFHQ
Wikiart Portraits



Datasets

We used different datasets

- **Style**

[WikiArt](#) (default): paintings in different artstyles

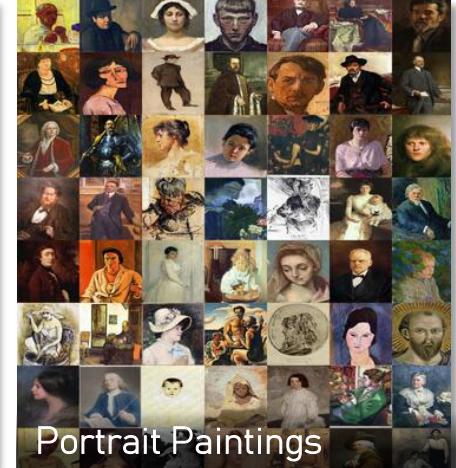
[LAION-Aesthetics](#): web images, including web artworks

[Portrait Paintings](#): subset of WikiArt with paintings of portraits

- **Content**

[COCO 2017](#) (default): pictures of generic objects

[Flickr-Faces-HQ](#): pictures of faces



Loss functions and distance metrics

Metrics among the VGG layers

- **MSE** (default):

$$MSE(y_{pred}, y_{true}) = \frac{1}{N} \sum_{i=1}^N (y_{true_i} - y_{pred_i})^2$$

- **L1**

More considerate of outliers.

$$L1(y_{pred}, y_{true}) = \frac{1}{N} \sum_{i=1}^N |y_{true_i} - y_{pred_i}|.$$

- **Cosine similarity:**

We got NaN loss during training.

$$S_{cos}(Y_{pred}, Y_{true}) = \frac{Y_{pred} \cdot Y_{true}}{\|Y_{pred}\| \|Y_{true}\|}$$

Similarity metrics for the computed image

- **Frechet Inception Distance:** measures the quality of the generated images.

It didn't suit our task.

Regularization

- **Total variation**

Pushes down the amount of varying pixels in the image, acting as a denoiser

$$TV(y) = \sum_{i,j} |y_{i+1,j} - y_{i,j}| + |y_{i,j} - y_{i,j+1}|$$

Optimizations

- **LPIPS**

use VGG weights fine-tuned for perceptual tasks

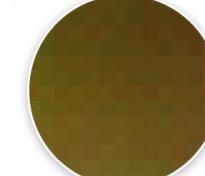
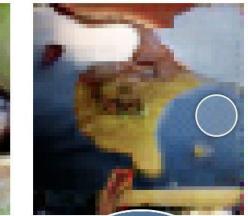
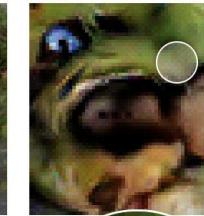
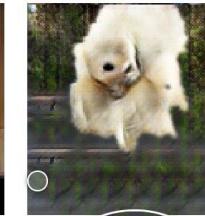
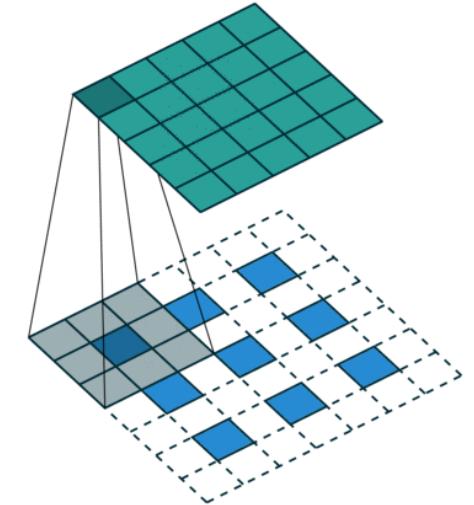
Transposed Convolution

- $K \times K$ parametrized filter is used for upsampling the image

$$H_{out} = (H_{in}-1) \times \text{stride}[0] - 2 \times \text{padding}[0] + \text{dilation}[0] \times (\text{kernel_size}[0]-1) + \text{output_padding}[0] + 1$$

$$W_{out} = (W_{in}-1) \times \text{stride}[1] - 2 \times \text{padding}[1] + \text{dilation}[1] \times (\text{kernel_size}[1]-1) + \text{output_padding}[1] + 1$$

- Problem: checkerboard artifacts (many convolutional layers required to smooth it out)



Radford等 , 2015 [1]

Salimans等 , 2016 [2]

Donahue等 , 2016 [3]

Dumoulin等 , 2016 [4]

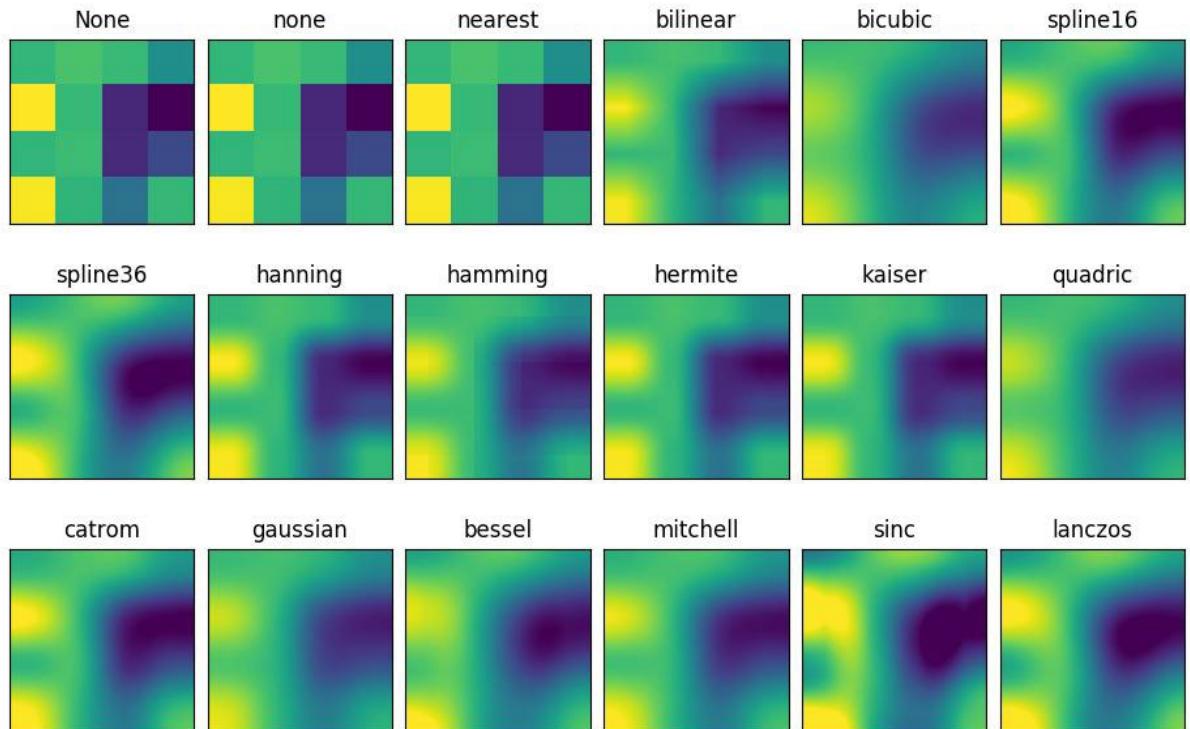
Alternative

Replace the transpose convolution with:

- **Upscaling** with bicubic interpolation
- **Convolution** with a big filter (15×15)

To enhance the edges

Big filter size to support high-res images



Hyperparameters Tuning

- **Content weight:** {0.5, 0.75, 1.0}
To give the model more freedom to edit the content
- **Style weight:** {3.0, 6.0, 9.0}
To apply the style more
- **Total Variation weight:** { $1e^{-5}$, $4e^{-5}$, }
To denoise the image
- **Style Signal Contrastive weight:** {3.0, 6.0}
To avoid collapse to NaN

For more details: [Weights & Biases Report](#)



The web app

```
355 #access {  
356   display: inline-block;  
357   height: 69px;  
358   float: right;  
359   margin: 11px 28px 0px 0px;  
360   max-width: 800px;  
361 }  
  
362 #access ul {  
363   font-size: 13px;  
364   list-style: none;  
365   margin: 0 0 0 -0.8125em;  
366   padding-left: 0;  
367   z-index: 99999;  
368   text-align: right;  
369 }  
  
370 #access li {  
371   display: inline-block;  
372   text-align: left;  
373 }  
374  
375 #access li a {  
376   color: inherit;  
377   text-decoration: none;  
378 }
```



About the app

Why an application?

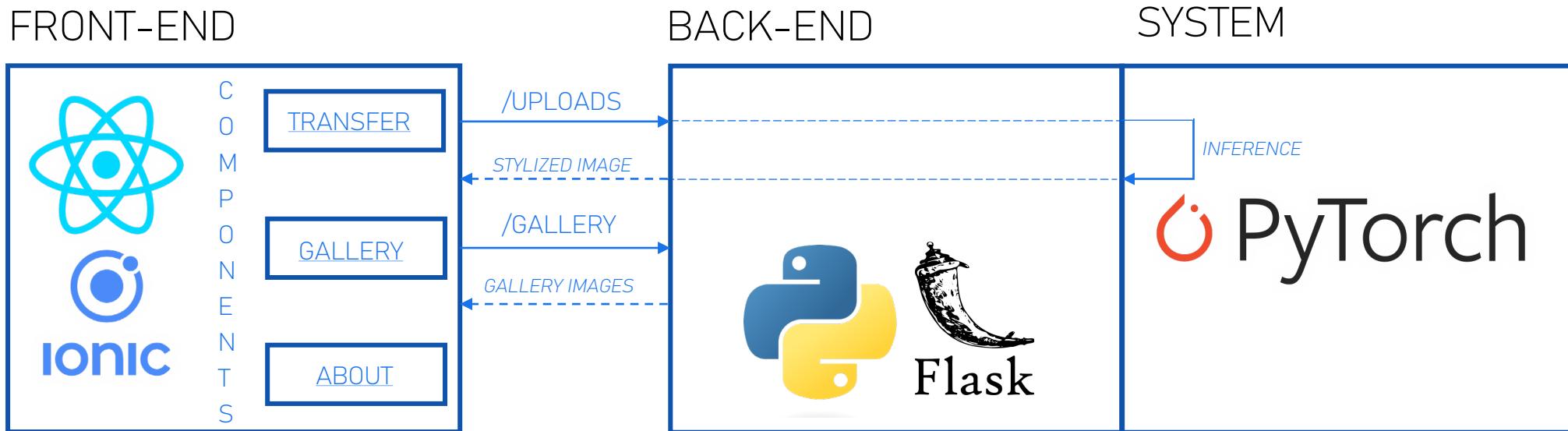
- Easier testing through UIs
- Provide end-users with an easy and customizable tool to edit their images
- Allow end-users to choose and understand differences between models and images

Client-server Application vs Pure Mobile Application

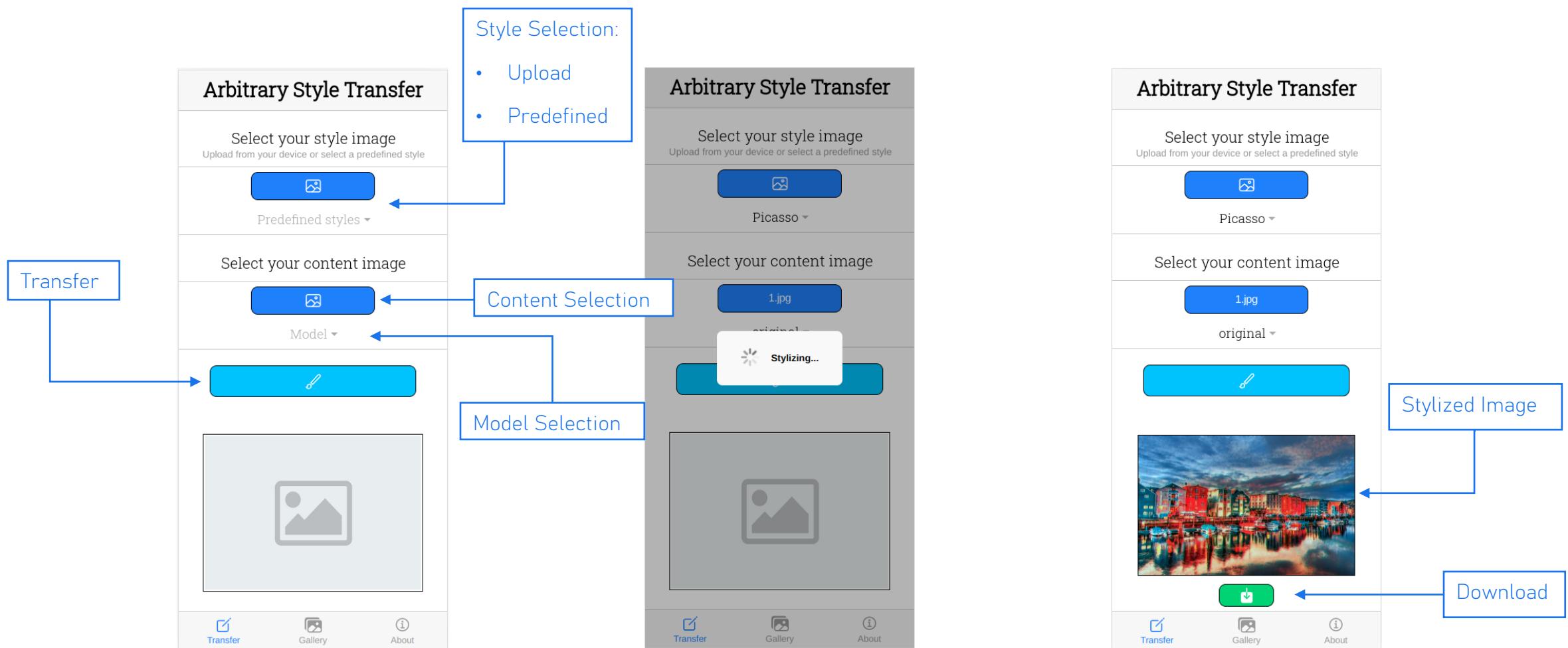
Use-Cases:

- Style Transfer (with relative customization)
- Gallery (with relative information associated to images)
- Source Code access

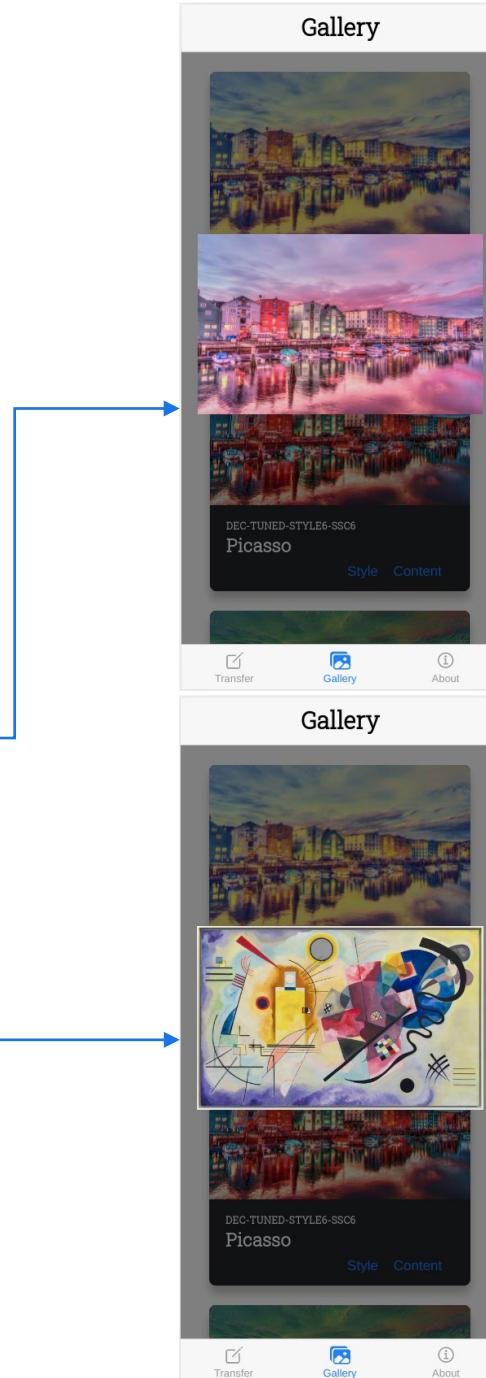
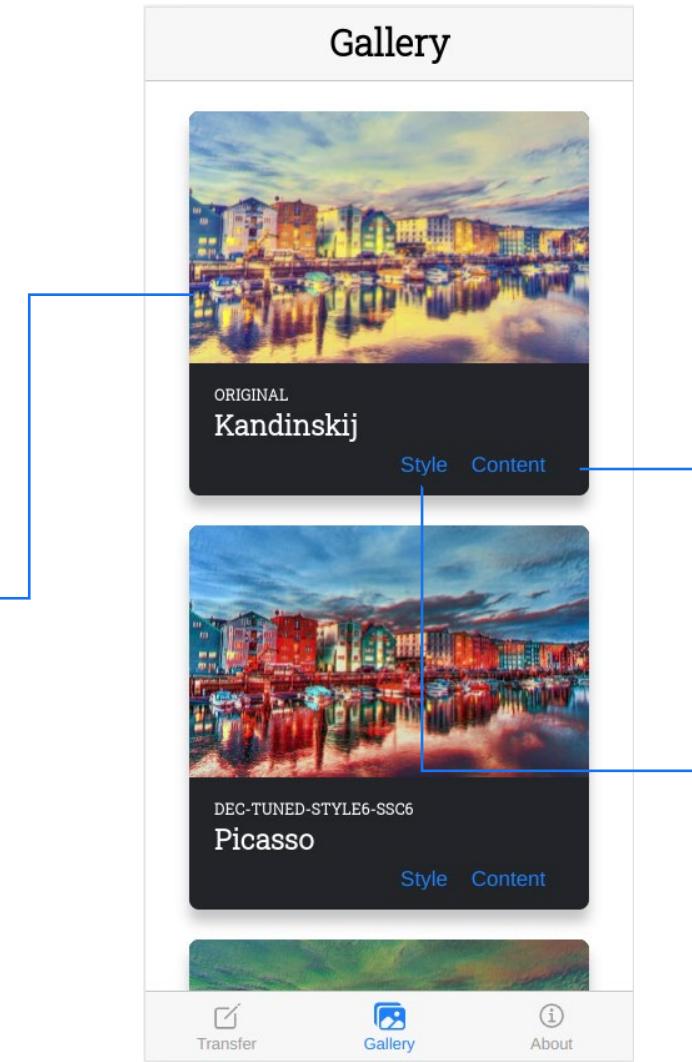
App Architecture



Demo: Style Transfer



Demo: Gallery



Demo: About

[Link to our GitHub project](#)

About the project

loss. MicroAST is 5-73 times smaller and 6-18 times faster than the state of the art, for the first time enabling super-fast (about 0.5 seconds) arbitrary style transfer at 4K ultra-resolutions.

Our Works

1. **Code Refactoring with new libraries:**
 - PyTorch Lightning
 - Torchmetrics
 - Weights & Biases
2. **Performance improvements through different experiments:**
 - Architecture
 - Loss functions
3. **Fine tuning of the model for two different tasks:**
 - Generic images
 - Images with faces
4. **Application development for testing the different models**

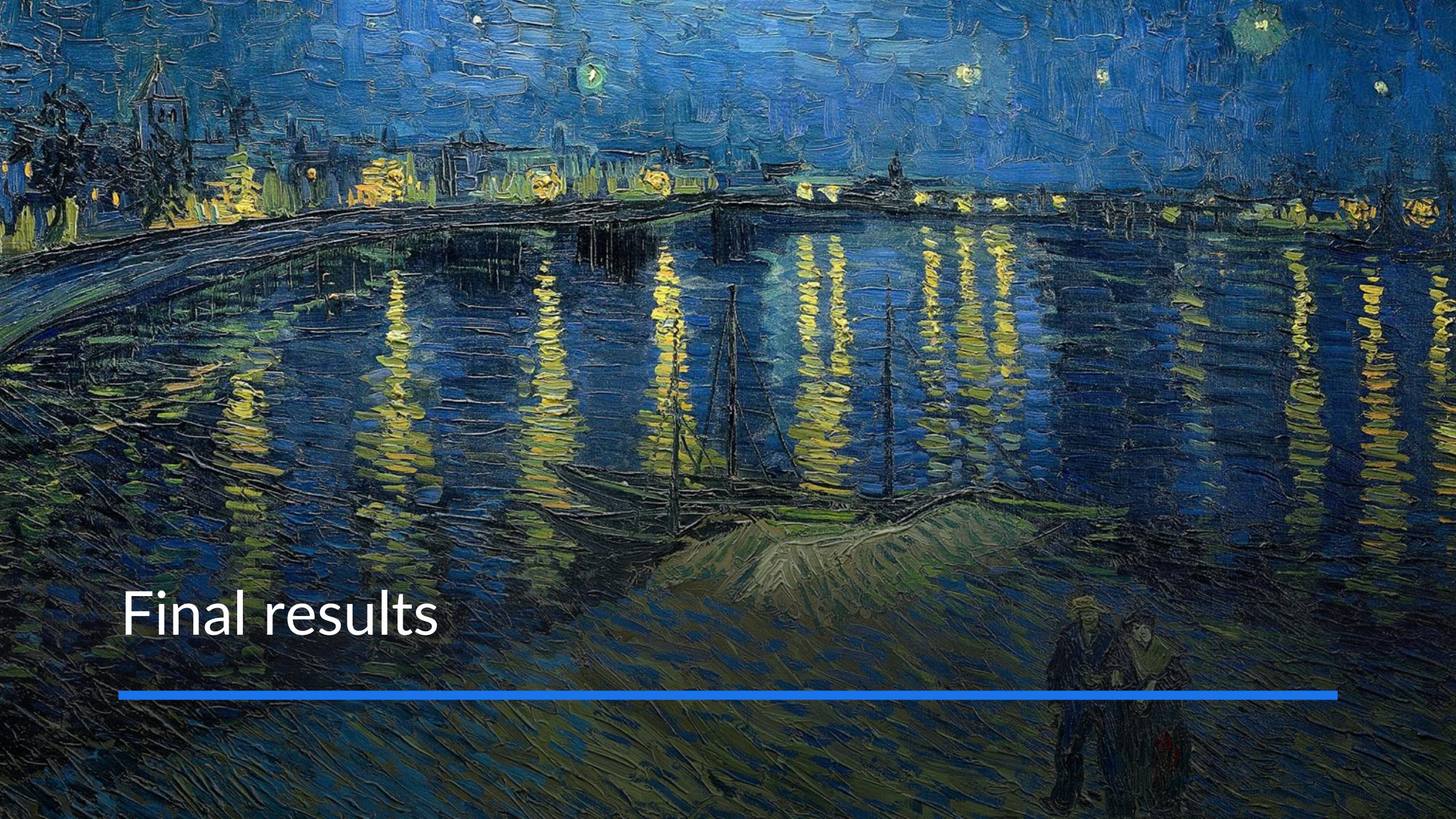
Credits

Antonio D'Orazio - 1967788

Robert Adrian Minut - 1942806

Giacomo Zarba Meli - 1807439



A reproduction of Vincent van Gogh's painting 'The Starry Night' is displayed against a dark blue background. The painting depicts a town at night with a church steeple and a bridge over a river. The sky is filled with stars and a crescent moon. The water reflects the lights from the town and the sky. In the foreground, two figures are walking along a path.

Final results

Generic



Original



CoCo/WikiArt + Decoder Tuned + TV



CoCo/WikiArt + Decoder Tuned



Generic



Original



CoCo/WikiArt + Decoder Tuned + TV



CoCo/WikiArt + Decoder Tuned



- *total variation regularization
- using MSE unless specified

Faces



Original



FFHQ/LAION*

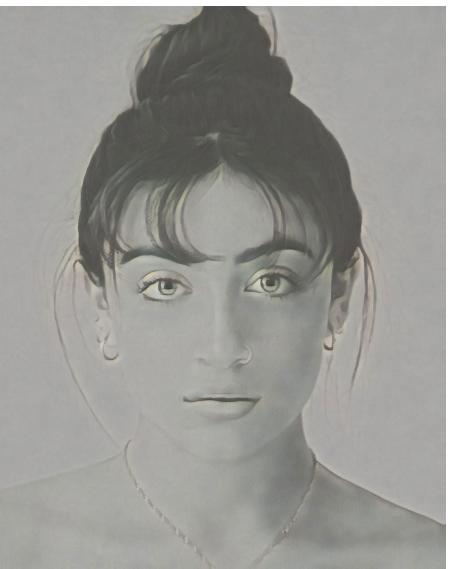


CoCo/WikiArt* +
Finetune FFHQ/LAION*



CoCo/WikiArt* +
Finetune WikiArt Portraits*

CoCo/WikiArt



CoCo/WikiArt*



- *total variation regularization
- using MSE unless specified

Faces



Original



CoCo/WikiArt* + Finetune
FFHQ/WikiArt Portraits*, L1, Style=6



CoCo/WikiArt* + Finetune
FFHQ/LAION*, L1, Style=9, Content=0.75

FFHQ/WikiArt Portraits*



CoCo/WikiArt Portraits*



- *total variation regularization
- using MSE unless specified

Faces



Original



FFHQ/LAION*

CoCo/WikiArt* +
Finetune FFHQ/LAION*



CoCo/WikiArt



CoCo/WikiArt*

CoCo/WikiArt* +
Finetune WikiArt Portraits*



- *total variation regularization
- using MSE unless specified

Faces



Original



CoCo/WikiArt* + Finetune
FFHQ/WikiArt Portraits*, L1, Style=6

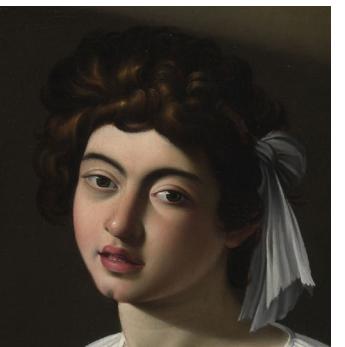
CoCo/WikiArt* + Finetune
FFHQ/LAION*, L1, Style=9, Content=0.75



FFHQ/WikiArt Portraits*



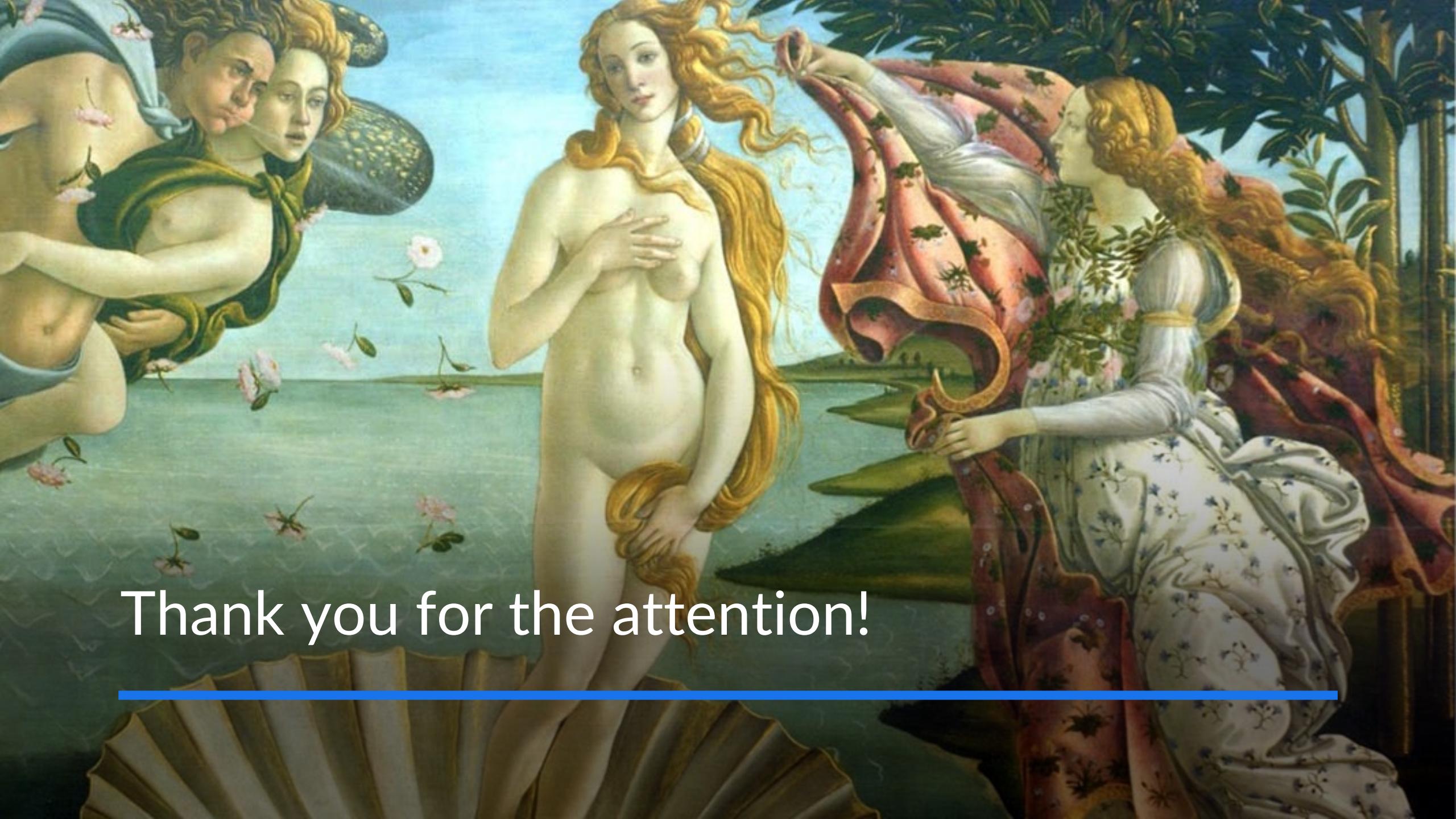
CoCo/WikiArt Portraits*



References

- [MicroAST: Towards Super-Fast Ultra-Resolution Arbitrary Style Transfer](#)
- [How to get beautiful results with neural style transfer](#)
- [Winning at loss functions: 2 important loss functions in computer vision](#)
- [Experiments on different loss configurations for style transfer](#)
- [Deep image quality assessment](#)
- [The Unreasonable Effectiveness of Deep Features as a Perceptual Metric \(arxiv.org\)](#)



A reproduction of Sandro Botticelli's painting "The Birth of Venus". The central figure is Venus, standing on a shell and emerging from the sea, her body glistening with water. She has long, flowing blonde hair and is looking towards the right. To her left, Cupid is flying on a pink dolphin, holding a bow and arrow. To her right, a woman (possibly a nymph or a personification of Nature) sits on a rock, holding a cornucopia overflowing with fruit and flowers. The background features a landscape with rolling hills and a clear blue sky.

Thank you for the attention!
