# Image style transfer with Facial preservation

Group22: 109550013林彥宇　109550135范恩宇

## 1. Introduction

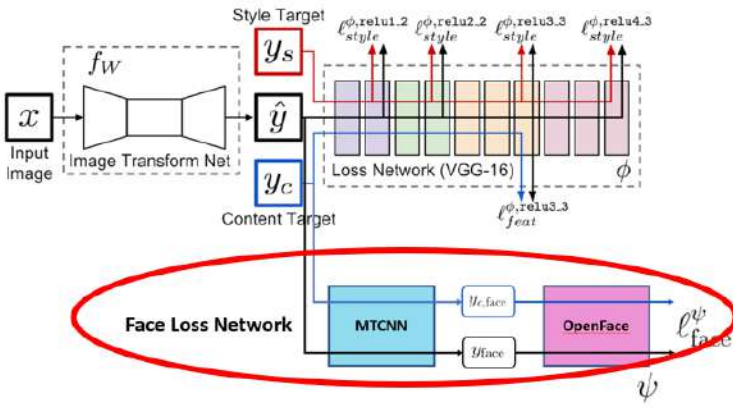
Image style transfer re-renders images in styles for multiple applications including art and education, often implemented by non-photorealistic rendering, optimization via Neural Style Transfer, and learning fast style transformations through deep learning. In this project, we try to use and train CNNs to change some styles for images.

Videos are made of frames, so we think it's possible to do style transfer on them. Besides, we want to try stylizing an image while preserving enough facial features by proposing an attention mechanism for facial focus, which is an extension upon the model of "Perceptual losses for real-time style transfer and super-resolution" by Johnson et al., hope to account for facial differences in stylized images.

## 2. Method & Extension

We train a feed-forward CNN to learn style transfer using perceptual loss, following Johnson et al's "Perceptual losses for real-time style transfer and super-resolution". The original model consists of two networks, an image transformation network stylizing input images, and a loss network computing perceptual loss from activation layers.

Because the original method of Johnson et al. took no special attention to particular content of the image, we define a specialized face loss network. Through this, we can penalize changes in areas around faces within the original image.



Our stylizing model is trained on COCO2017 dataset, with 118 thousand images rendered for 2 epochs, with style images below.

### 2.1 Perceptual Loss

Perceptual loss uses the perceptual features of a pre-trained "loss" network , to compute perceptual difference between two images. In our case, the pretrained loss network is an image classifier trained on an image classification dataset(we use the 16-layer VGG network pretrained on ImageNet).

Activation layers of the loss network represent different perceptual knowledge of the image. Earlier ones quantify style elements of the image, like color and texture. Later ones tell about deeper contents of the image.

#### 2.1.1 Content loss

With the loss network, we can compute the loss in content of stylized image as being the difference in activation layers. For a stylized image y and original content image yc, the content loss is defined as this equation done by Johnson et al we mentioned previously, phi j (x) means the jth activation layer (of shape Cj x Hj x Wj ) of the network phi for input x.

$$\ell_{\text{content}}^{\phi,j}(y, y_c) = \frac{1}{C_j H_j W_j} \|\phi_j(y) - \phi_j(y_c)\|_2^2$$

#### 2.1.2 Style loss

Compute the loss in style of stylized image as being the difference in activation layers. Through Gram matrix and formula from Gatys et al., the style loss of a stylized image y and style image ys, is defined as the formula below.

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}$$

$$\ell_{\text{style}}^{\phi,j}(y, y_s) = \|G_j^{\phi}(y) - G_j^{\phi}(y_s)\|_F^2$$

For multiple activation layers, we take the sum of the individual style losses

#### 2.1.3 Total Variation Regularization

Johnson et al. also suggest using total variation regularization to encourage spatial smoothness. By using perceptual loss and combining the three parts we just mentioned, we can define a loss function for our image transformation network to train, then we can do the style transfer.

## 2.2 Extension 1: Real-time Video

We focus on video frames and functions for reading and writing video files with matched properties. First, set attributes like frame count, fps, shapes, then prepare a buffer for batch processing. During the styling process, first read frames in batches from input video, then convert them into PyTorch tensors and yield them(we manage batch size dynamically, to ensure proper handling for cases where frames remain less than the batch size). When the reading process is completed, apply a stylizing model to the content.

For the writing part, we take batches of frames as PyTorch tensors and convert each frame to a NumPy array. After adjusting the dimensions appropriately, write each frame to the output video file.

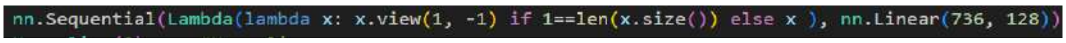
## 2.3 Extension 2: Facial Preservation

Base perceptual loss introduced by Johnson et al only preserves content on a global level, so we add an additional loss term to account for differences in facial features between original and stylized image. By including this loss term, we can avoid making facial features like nose or mouth covered by style features and become unable to identify.

To quantify the difference in facial features between the original img and stylized image y, we use face recognition. First find faces in original images with Multi-task Cascaded Convolutional Networks (MTCNN), which produces bounding boxes for faces. By these bounding boxes, patches are extracted and resized (96 × 96).

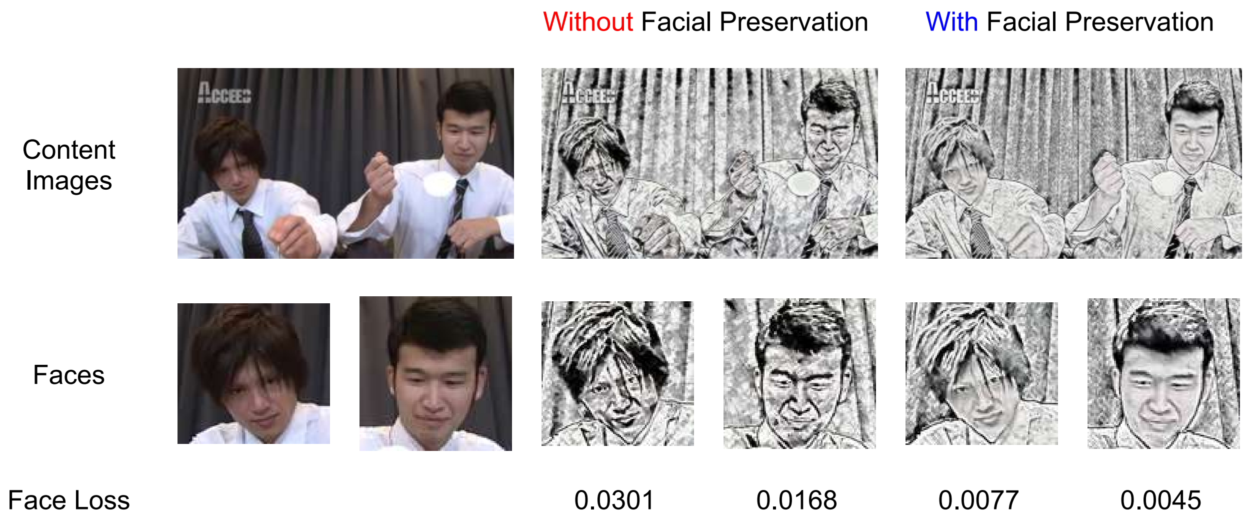
Then with these resized patches and OpenFace we found on github that can extract facial features , we can compute a face descriptor embedded on a 128-dimension hyper-sphere, and then take distance between 2 face descriptors of original & stylized image as a measure of similarity, for defining face loss.

The face loss network should reduce the spatial dimensions while increasing abstraction as it goes deeper. In addition, for the features we extract for face preservation, they should represent more abstract concepts derived from the input image data through multiple layers of processing. So eventually, we choose high-level features from the OpenFace model, especially from a linear layer that transforms an input vector of 736 features into an output vector of 128 features with a learned weight matrix and bias vector.



```
nn.Sequential(Lambda(lambda x: x.view(1, -1) if 1==len(x.size()) else x ), nn.Linear(736, 128))
```

Take manga style for example, we trained two models for each image, one without and one with facial preservation. Images here are results we get. Faces row shows face patches found by MTCNN, last row is the face loss between stylized and original faces. In most cases, facial preservation decreases face loss.

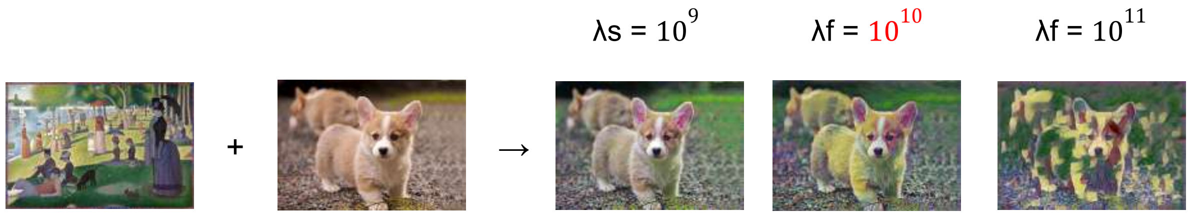


In the training process, models without facial preservation usually take 12~16 hours, but those with facial preservation are often more than a day, so we eventually just trained 6 models without facial preservation and 3 with facial preservation.

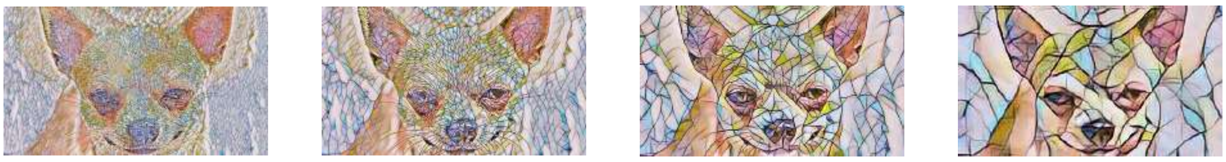
## 3. Experiments

### 3.1 Stylizing

We tried various style weights to improve the quality, and we found that making style loss weight = $10^{10}$, content loss weight = $10^{5}$, and total variation regularization weight = $10^{-6}$ produce probably the best results.

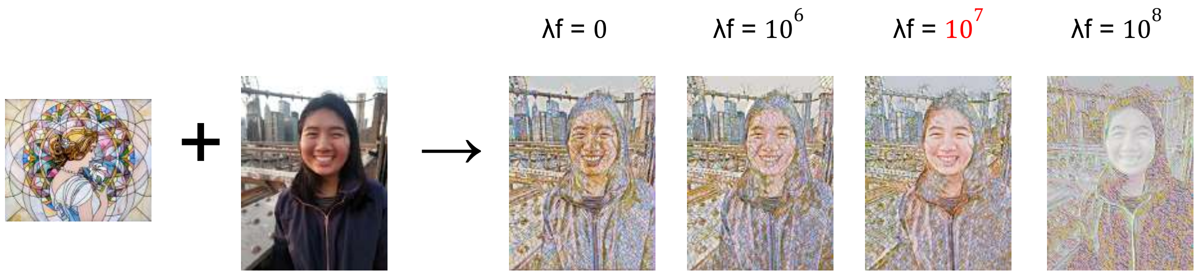


Resolution of the input images also affects the results of stylization.



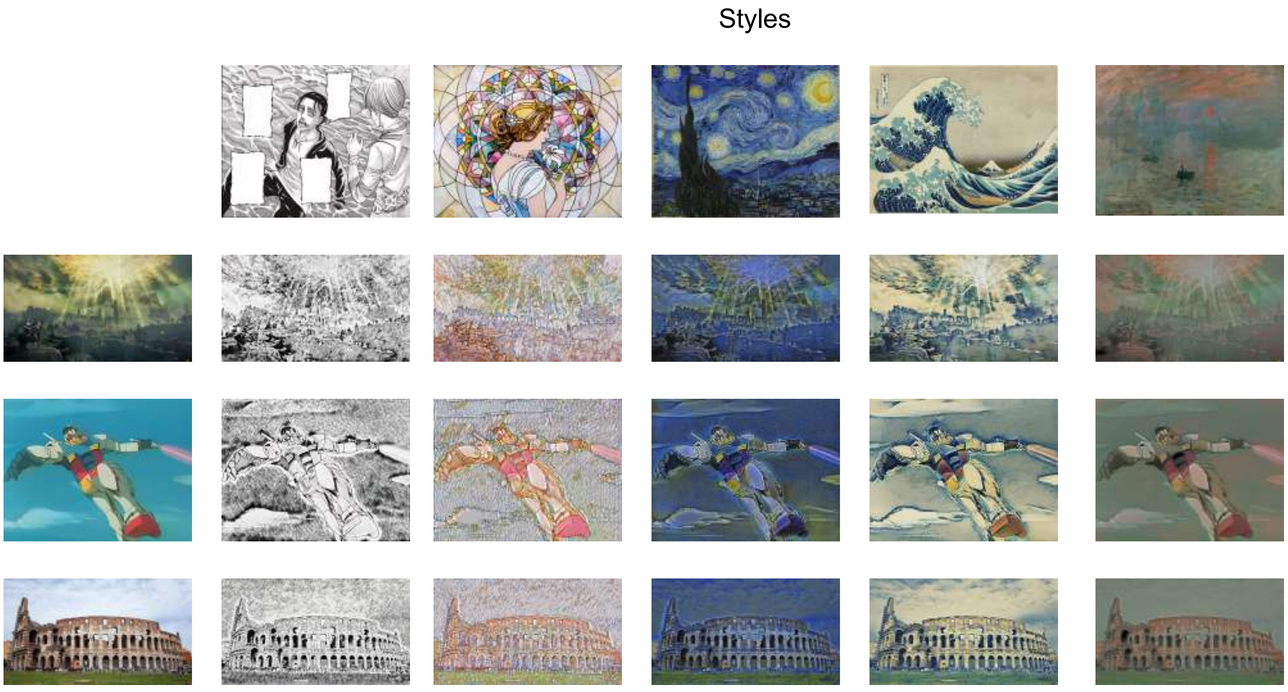
## 3.2 Facial Preservation

We tried different weights and found that setting face loss to $10^7$ gives the best result for preserving clear facial features, while also showing enough applied style.



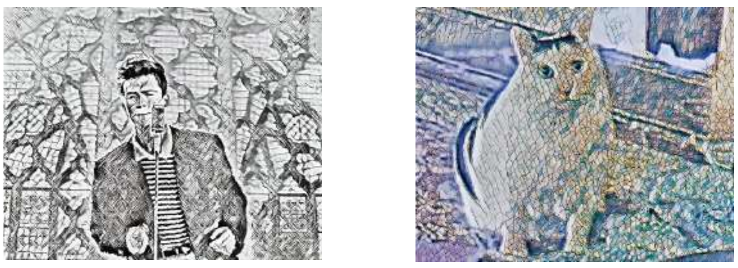
## 4. Results

### 4.1 Stylizing Results(5 styles tested)

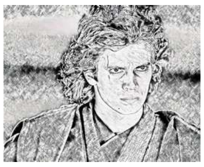
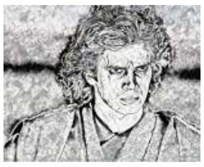
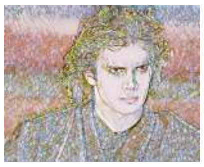
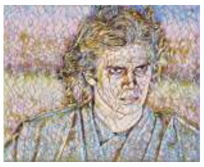
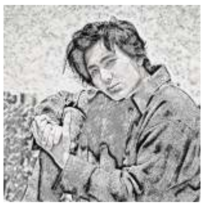
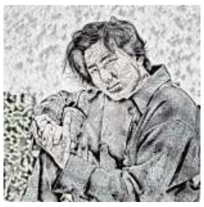
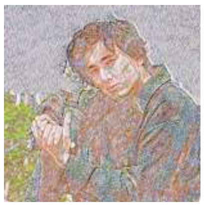
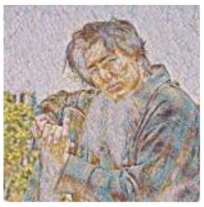
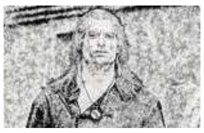
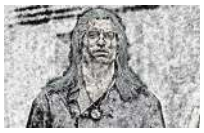
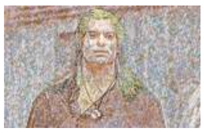
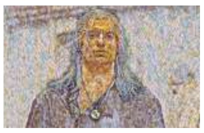


### 4.2 Video stylizing results

Two different styles on two videos, but it doesn't seem as good as images.



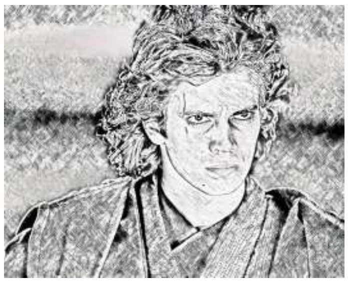
### 4.3 Facial preservation(FP) results

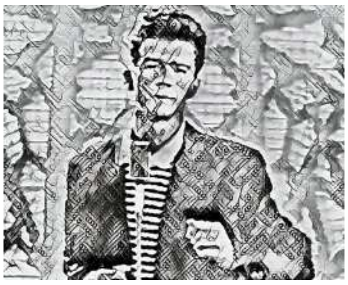
| Original Image | manga,with FP | manga,without FP | mosaic,with FP | mosaic,without FP |
|---|---|---|---|---|
| |  |  |  |  |
| |  |  |  |  |
| |  |  |  |  |

## 5. Compaison & Discussion

### 5.1 Different Style Effect: Image & Video

According to experiments and results, we found that facial preservation results in a noticeable difference when applying styles to images and videos. Results in videos tend to be worse, which might be due to the need for temporal consistency in videos.

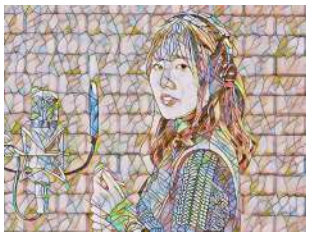

facial preservation on image

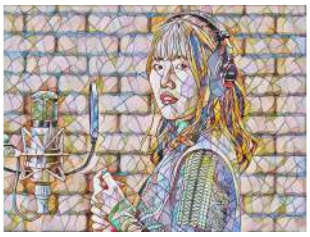

facial preservation on video

### 5.2 With & without Facial Preservation

Besides, facial preservation does maintain the quality of facial features. In images, facial features with facial preservation are clearer, while those without facial preservation are often affected by the style and harder to be identified. In addition, parts that aren't faces in images sometimes look more harmonious if facial preservation is applied.

original image


with facial preservation


without facial preservation

## 6. Conclusions

### 6.1 Achievements

First, with experiments about different style images, style weights, and resolution dependence, the style transfer system is proved effective.

Second, since videos are actually a series of frames, stylizing each of them does change the style of videos, but it needs some modification.

Most importantly, realizing facial preservation not only can keep facial features in most images, but also make images apply some styles more properly, which is actually an unexpected discovery.

### 6.2 Limitations

Facial preservation sometimes cannot work thoroughly, especially when there is hair or stubble covering the face. Also, style features are resolution dependent while better resolution leads to better stylization with facial preservation. In addition, facial preservation doesn't work on videos well, maybe it's because of temporal consistency.

### 6.3 Future Work

Train a new model on a dedicated face dataset for better facial preservation to avoid face detection/descriptor bottlenecks, then we should be able to improve facial preservation with face loss and make this technique work better on videos.

## Libraries

- Pytorch
- Numpy
- PIL

## Open Source Codes

https://github.com/TropComplique/mtcnn-pytorch
https://github.com/TwoBranchDracaena/OpenFace-PyTorch
https://github.com/pytorch/examples/tree/main/fast_neural_style

## References

1. D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stylebank: An Explicit representation for neural image style transfer.
2. L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style.
3. J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution.
4. J. Wang, Y. Yuan, and G. Yu. Face attention network: An effective face detector for the occluded faces
5. H. Zhang and K. J. Dana. Multi-style generative network for real-time transfer
6. K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks.

## Contribution

We have equal contribution for coding, presentation, and report.