Image-to-Sketch with Color-Preservation Style Transfer

Introduction

The purpose of this project is to convert images into pencil sketches. Numerous methods have been attempted, including OpenCV's built-in functions and neural style transfer (NST). In order to transfer style while maintaining colours, a pre-trained VGG network was combined with a simple linear method. The performance of each method will be discussed in detail.

Methodology

Approach 1: Combination of different basic cv2 functions

Using the cvtColor function, we first convert the image to grayscale. The bitwise_not() function is then used to invert the pixel intensities. For smoothing inverted images, OpenCV's GaussianBlur() function is implemented with kernel size (21, 21) and zero standard deviation. The image is finally normalized using the divide() function.

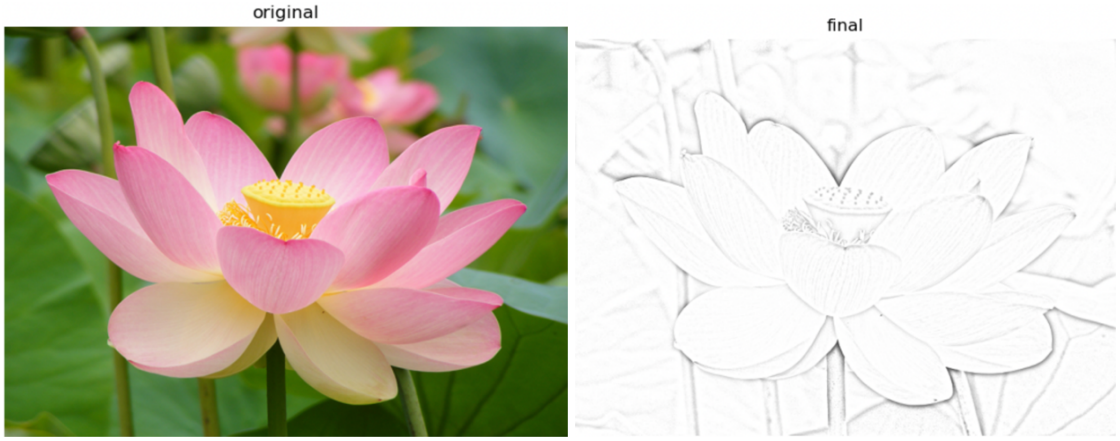

Figure 1. Approach 1 result

This approach meets the bare minimum requirements for image-to-sketch applications. Colors and small details of the input image are not preserved, however.

Approach 2: OpenCV's pencilSketch() function

The input image is converted to RGB and passed to the pencilSketch() function. A grayscale sketch image and a color sketch image are returned by the function, which takes 4 parameters: the input image, sigma_s, sigma_r, and shade_factor (figure 2). It is sigma_s that determines the size of the neighborhood, whereas sigma_r determines how different colors within the neighborhood will be averaged. The larger sigma_r, the larger the areas of constant color [1].

Figure 2. Approach 2 result

This built-in function performs excellently in preserving colors and rendering vivid pencil strokes.

Approach 3: Style Transfer

The NST takes two images as input: content image (C) whose content we want to retain and style image (S) whose artistic characteristics are transferred onto (C). NST has been widely proposed in Image-to-Sketch literature [2], [3]. The main issue with this approach is that it cannot effectively preserve colors of the content image. To address this, the luminance-only transfer method demonstrated in [4], [5] is employed.

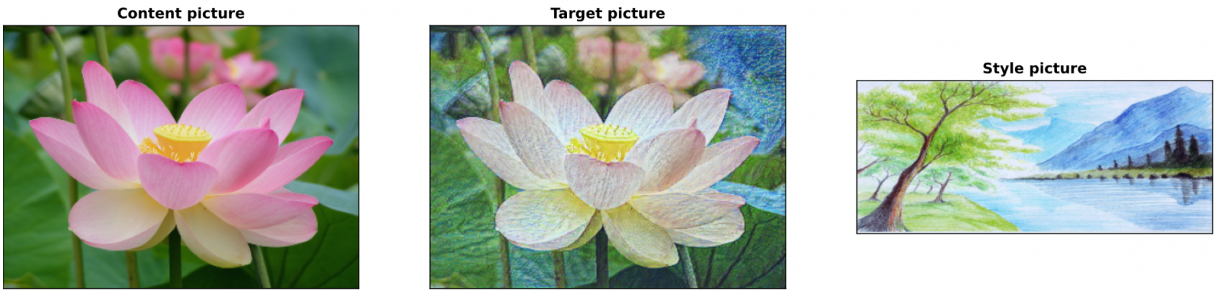

Figure 3. NST without luminance transfer

Luminance-only transfer: This function extracts luminosity characteristics L_s, L_c of the style and content images. Let μ_s and μ_c be the mean luminance of the two images, σ_s and σ_c be their standard deviations. Then each luminance pixel in the style image is updated as

$$L_{s'} = \frac{\sigma_c}{\sigma_s}(L_s - \mu_s) + \mu_c$$

The style image is processed by the luminance-only transfer function to match the colors of the content image. The resultant image is then used as input for the NST.

The edges and details of the style image are enhanced using cv2 before it is passed to the luminance-only transfer function for better final results.

Using the lotus content image displayed above, the original and transformed style images are as follows:

Figure 4. Luminance transfer result

NST: The first step is to import and freeze a pre-trained VGG19. A trainable target image is created using random numbers (set requires_grad = true). Then the content and style images are transformed and then put through the Conv2D layers of the CNN to extract feature maps.

For content matching, the mean squared error (MSE) between the target activation maps and the content activation maps is calculated. For style matching, the gram matrices of the target activation maps and style activation maps for different layers are computed. This gives the style MSE. Adding the content and style losses, we have the loss that the model uses during backpropagation.

Additionally, as the numerical values of the content MSE and style MSE are very different, the style MSE needs to be scaled. Moreover, as we go deeper into the network, the later Conv2D layers contain more abstract characteristics. Thus, for style matching, higher weights will be given to earlier layers to focus more on finer details and less on the larger scale patterns.

The content, resulting target and style images are as follows:

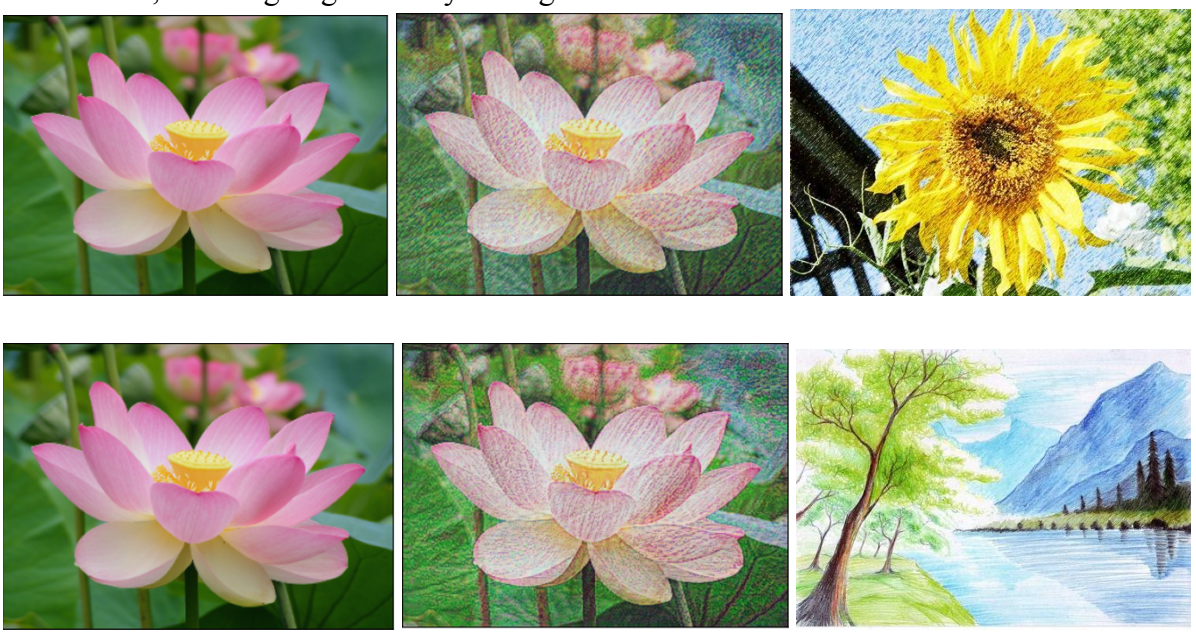

Figure 5: NST with luminance transfer

This approach yields reasonable results. However, its performance depends on our ability to hand-pick the appropriate style image.

Conclusion

While approach 2 is the most effective method in converting images to sketches, approach 3 is applicable to different styles of sketching. Approach 3 can be improved by, for example, using a deeper, more efficient neural network or choosing more suitable style images.

Citations:

[1] Mallick, Satya. “Pencilsketch.” LearnOpenCV, 21 Mar. 2015, https://learnopencv.com/tag/pencilsketch/.
[2] P. Chandran, G. Zoss, P. Gotardo, M. Gross and D. Bradley, "Adaptive Convolutions for Structure-Aware Style Transfer," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 7968-7977, doi: 10.1109/CVPR46437.2021.00788.
[3] W. Zhang, G. Li, H. Ma and Y. Yu, "Automatic Color Sketch Generation Using Deep Style Transfer," in IEEE Computer Graphics and Applications, vol. 39, no. 2, pp. 26-37, 1 March-April 2019, doi: 10.1109/MCG.2019.2899089.

[4] Gatys, Leon & Bethge, Matthias & Hertzmann, Aaron & Shechtman, Eli. (2016). Preserving Color in Neural Artistic Style Transfer.
[5] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann and E. Shechtman, "Controlling Perceptual Factors in Neural Style Transfer," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 3730-3738, doi: 10.1109/CVPR.2017.397.