10.9日论文

1. 学习将纹理显著性自适应注意纳入图像卡通化（Learning to Incorporate Texture Saliency Adaptive Attention to Image Cartoonization）

icml2022

该论文提出了一个新颖的图像动漫风格迁移模型，建立了图像级与区域级并行的对抗学习，区域级分支将对抗学习约束在卡通图像纹理显著的区域，以更好地感知和传递卡通纹理特征。这种方法可以高效地深入挖掘动漫图像的纹理特征和表征分布，在多个数据集，尤其是高分辨率数据集上实现了更为显著、生动的风格渲染效果。该研究为卡通风格迁移的研究开创了新的思路。

摘要——从无监督图像到图像翻译的角度来看，图像卡通最近由生成对抗网络（GANs）主导，其中一个固有的挑战是精确捕捉和充分转移特征的卡通风格（如清晰的边缘、平滑的色彩阴影、抽象的精细结构等）。

Existing advanced models try toenhance cartoonization effect by learning to promote edges adversarially, introducing style transfer loss, or learning to align stylefrom multiple representation space.

本文证明，仅需基本的对抗性损失，就可以实现更清晰、生动的制图效果。观察到卡通风格在卡通纹理显著的局部图像区域更加明显，我们构建了一个与正常图像级并行的区域级对抗性学习分支，它限制了对卡通纹理显著的局部斑块的对抗性学习，以更好地感知和传递卡通纹理特征。

CTSs模块： cartoon texture salience sampler

为此，提出了一种新的卡通纹理显著性采样器（CTSS）模块，从训练数据中动态采样卡通纹理显著性斑块。通过大量的实验，我们证明了对抗性学习中的纹理显著性自适应注意作为图像卡通化中相关方法缺失的组成部分，在促进和增强图像卡通风格化方面具有重要意义，特别是对于高分辨率的输入图片。

随后介绍针对图像卡通化的已有方法：图像平滑、颜色量化；（iii）边缘增强；（虽然它们成功地模仿了一些重要的卡通特征，但它们缺乏数据驱动的学习能力来更深入地捕捉卡通风格。）

(a) Original Picture (b) Our Result

然后简单叙述GAN参与图像卡通化的原理：

它可以表述为一个无监督的图像到图像转换问题，其目标是从自然图像的源域X→Y学习保留内容的图像转换映射X到卡通图像的目标域Y。一般的框架是通过对抗性学习将生成图像的风格分布与目标域真实漫画的风格分布对齐，同时约束输入照片和生成结果之间的感知内容一致性，避免内容不匹配。然而，很难产生具有足够显著的卡通特征的结果，因此一些先进的方法在一般框架的基础上进一步提高了卡通化效果。

经典之作CartoonGAN：提出了一种新的**促进边缘的对抗性损失**来突出典型的边缘清晰度的卡通特征。该损失函数强制鉴别器不仅区分真实的卡通图像和合成的图像，而且区分边缘平滑的卡通图像，从而引导生成器产生更清晰的边缘，欺骗鉴别器。

AnimeGAN [7]将Gram损失[8]，一种经典的基于纹理描述符的风格损失，引入到GAN框架中，以增强学习卡通纹理模式。然而，它对加强卡通纹理转移的作用仍不那么明显。最近，提出了一种白盒图像卡通化框架[9]。它将图像分解为多个表示，并学习在每个表示的流形中对齐样式。该方法提出了一种自适应着色算法，生成图像颜色分割地图，模拟卡通图像的稀疏色块，带来视觉上的卡通风格。然而，生成的结果的卡通抽象性和生动度仍然不那么突出，特别是对于高分辨率的输入图像。建议下次这么写的时候标出来哪里不好了！

上述模型采用不同的方法来弥补在充分转移卡通风格时基本对抗性损失的局限性。然而，我们认为弱程式化问题不是由于对抗性损失的能力本身，而是由于突出的卡通纹理特征的非全局分布。例如，清晰的边缘通常分布在局部区域，而不是整个图像，并且清晰的边缘的像素比例也很小。
因此，当在整个图像的尺度上进行基本的对抗性损失训练时，边缘清晰度的特征很容易被更明显的全局特征所压倒，如颜色阴影的平滑度。

这启发我们关注卡通纹理突出的局部区域，以便更好地感知和传递卡通纹理特征。

同样的关注图像补丁的局部分布的思想在许多其他计算机视觉领域也被广泛探索，如图像分类[10]、[11]、[12]、图像恢复[13]、[14]、[15]、目标检测[16]、[17]、[18]等。

本文作者的意见⇑

In this paper, we propose a compact and efficient adversarial learning framework with an image-level discriminator examining global cartoon styles like smooth shading and the overall tone, as well as a patch-level discriminator focusing on learning local cartoon texture pattern, i.e., the unique distribution of high- and low-frequency pixels around clear edges. To enhance transfer of cartoon texture pattern, we adaptively constrain patch-level adversarial learning on cartoon-texture-salient local image regions, for which a novel cartoon-texture-saliency-sampler (CTSS) module is proposed to dynamically extract image patches with most salient cartoon texture pattern from each mini-batch of training images. By incorporating

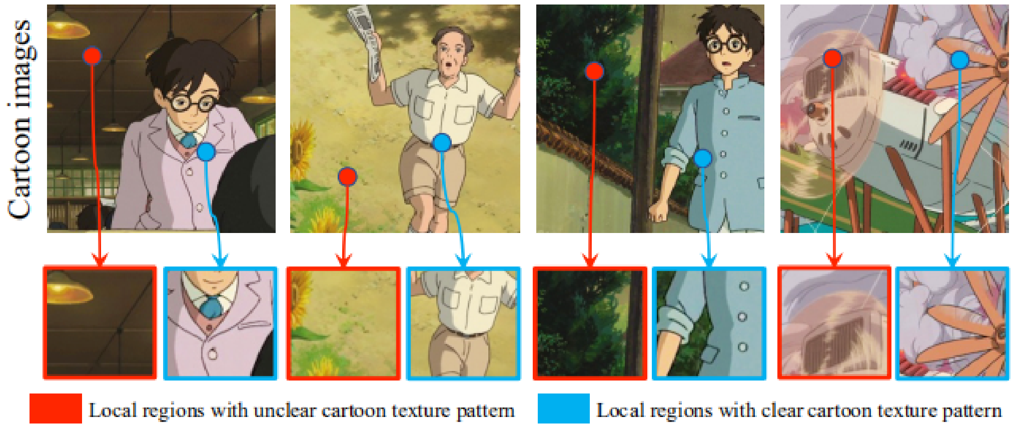
为了增强卡通纹理模式的传输，我们自适应地约束了卡通纹理的局部图像区域的补丁级对抗性学习，提出了一种新的卡通纹理采样器（CTSS）模块，从每个小批训练图像中动态提取具有最显著卡通纹理模式的图像块。

通过将纹理显著性自适应注意纳入对抗性学习，典型的卡通纹理特征被更充分地转移，产生更抽象、生动的卡通效果。我们的模型的示例结果如图1所示。我们的方法绕过了单独的边缘平滑数据准备阶段，使用额外的风格损失，以及复杂的表示提取过程，同时产生更突出的卡通效果，特别是对于大的输入图片。

随后是简单的综述：

1. 神经样式传输（NST）最初是被提出的一种基于在线优化的算法，通过最小化克损失[8]来迭代传输图像样式。然后，通过训练一个前馈网络[19]，[20]，将其转化为一个离线生成模型，以满足实时应用程序的需求。后来，人们努力将快速NST从单一风格扩展到多种风格的[21]，[22]，甚至是任意风格的[23]，[24]。除革兰氏损失外，还先后提出了各种风格损失，如MMD损失[25]、CNNMRF损失[26]、上下文损失[27]和松弛EMD损失[28]。这些损失函数适合于从单一图像传输低级纹理特征。相比之下，跨域样式转移方法使用对抗性损失来自动从风格上相似的图像集合中学习高级样式。通过GANs，风格转移问题得到了更多的应用，如字体风格转移[29]、[30]、绘画风格转移[31]、[32]、化妆风格转移[33]等。
2. 图像-图像转换是指图像从源域到目标域的转换，根据两个域的配对训练数据是否可用，可分为监督和无监督情况。对于监督问题，Pix2Pix [34]将条件GAN与图像级稀疏L1正则化相结合，可以很好地推广到图像超分辨率[35]、图像去噪[36]、语义图像合成[37]等许多应用中。对于后一种情况，CycleGAN [38]和UNIT [39]分别是基于周期一致性约束和共享潜在空间假设实现无监督图像平移的典型模型。然后，将StarGAN [40]和AttGAN [41]等方法通过将条件GAN与辅助域分类器相结合，将翻译从两个域扩展到多个域。此外，在多模态翻译[42]、[43]和少样本学习[44]领域也探讨了这个问题。

卡通图像有光滑的阴影和生动的色彩。除了这些代表整体像素分布的全局特征外，卡通图像最显著的特征是其独特的纹理模式，代表局部像素分布。具体来说，高频像素集中在边缘上，而低频像素则平滑地分布在边缘旁边。这种明显的高频和低频像素的分离明显不同于自然图像，其中高频和低频元素紧密地交织在一起。然而，如图3所示，



这种卡通纹理图案

Fig. 3. The typical cartoon texture pattern manifests clearly only in partial image regions with distinct edges.

只在边缘清晰的部分图像区域中明显出现，这意味着潜在的卡通纹理图案从边缘不同的局部区域的视角中比从整个图像的视角中更容易感知。因此，我们附加了一个补丁级学习分支，它自适应地对边缘不同的局部区域应用对抗性学习，以增强对卡通纹理模式的捕获。

作者想法来源

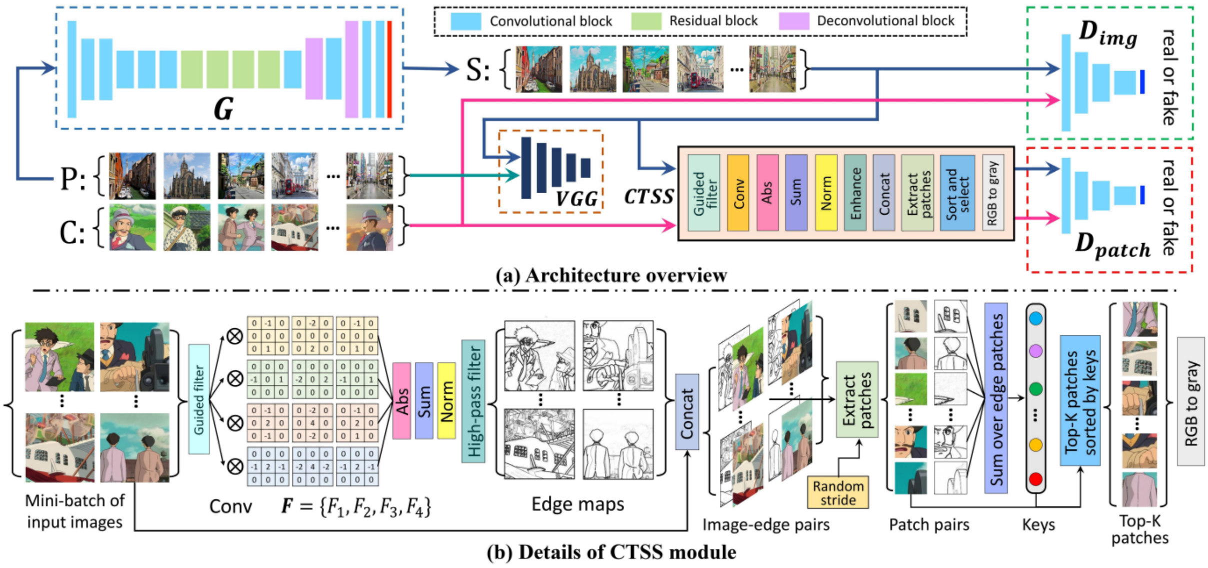


Fig. 2. The overall architecture of our model, as well as details of our proposed cartoon-texture-saliency-sampler (CTSS) module which adaptively extracts local image patches with most salient cartoon texture pattern from each mini-batch of input images.

A. Global and local adversarial learning

Let $\mathcal{P}$ denote the domain of real-world photos, $\mathcal{C}$ be the domain of cartoon images, $\mathcal{S}$ be the domain of synthesized results. The overall architecture of our adversarial learning framework is illustrated in Fig. 2 (a). In training phase, a mini-batch of natural photos $P = \{p_i\}_{i=1}^N \in \mathcal{P}$, and a mini-batch of cartoon images $C = \{c_i\}_{i=1}^N \in \mathcal{C}$ are sampled at each iteration, where N is the batch size. A generator G translates P into a mini-batch of synthesized cartoon images $S = G(P) = \{s_i\}_{i=1}^N \in \mathcal{S}$, which are differentiated from real cartoon images C by an image-level discriminator D_{img}. This forms the image-level adversarial learning branch for learning global holistic cartoon styles.

To better seize cartoon texture pattern that is more perceptible at edge-distinct local regions, we append a patch-level adversarial learning branch as a supplement to the image-level one. In this branch, a cartoon-texture-saliency-sampler (CTSS) module is proposed to constrain adversarial learning only on cartoon-texture-distinct local regions. As shown in Fig. 2 (a), the CTSS module takes cartoon mini-batch C and the synthesized mini-batch S as input, and outputs top-K edge-distinct local patches $C_{patch} = \{c_p^i\}_{i=1}^K$ and $S_{patch} = \{s_p^i\}_{i=1}^K$ from C and S respectively. A patch-level discriminator D_{patch} is built to distinguish S_{patch} from C_{patch}, forming the patch-level adversarial learning that reinforces transfer of cartoon texture pattern.

B. Cartoon-texture-saliency-sampler

Since the unique cartoon texture pattern is more visually prominent at edge-distinct local regions, our CTSS module adaptively samples local image patches with most distinct edges from each input mini-batch of images, the implementation details are illustrated in Fig. 2 (b). Taking an input mini-batch of cartoon images $C = \{c_i\}_{i=1}^N$ as example, CTSS starts with a guided-filter [45] sub-module $\mathcal{F}_{gf}$ for edge-preserving image smoothing, it uses each input image c_i itself as guide map, returns the smoothed image $\tilde{c}_i$ with many noise elements removed:

$$\tilde{c}_i = \mathcal{F}_{gf}(c_i, c_i), i = 1, ..., N. \tag{1}$$

Then, a convolutional layer is applied to extract coarse edge maps $E = \{e_i\}_{i=1}^N$, where e_i is the edge map of c_i. The convolutional layer has a constant kernel $\mathbf{F}$ composed of four filters $\{F_1, F_2, F_3, F_4\}$ as shown in Fig. 2 (b). The designed kernel $\mathbf{F}$ is specially suitable for cartoon edge extraction, and is essentially an improved Sobel operator. The coarse edge map is obtained by summing over the absolute value of the convolution result with each filter of $\mathbf{F}$, followed by Min-Max normalization to rescale to $[0-1]$:

$$e_i = Norm_{min\_max}(\sum_{k=1}^{4} |\tilde{c}_i \otimes F_k|), i = 1, ..., N, \tag{2}$$

where $\otimes$ denotes convolution. Eq. 2 can be efficiently implemented with a single-layer convolution with kernel $\mathbf{F}$ followed by channel-wise manipulations. Based on coarse edge maps E, the refined edge maps $\tilde{E} = \{\tilde{e}_i\}_{i=1}^N$ are obtained by applying a high-pass filter $h(\cdot)$ that enhances high-frequency pixels and suppresses low-frequency ones:

$$\tilde{e}_i = h(e_i) = 1 - 1/(1 + (e_i/d)^n), i = 1, ..., N, \tag{3}$$

where d and n are hyperparameters that determine threshold and sharpness of the high-pass filter $h(\cdot)$ respectively. Visualization of the final refined edge maps $\tilde{E}$ is shown in Fig. 4. The refined edge maps are used to adaptively guide attention to edge-distinct local image regions and extract corresponding image patches:

$$\{c_p^i, e_p^i\}_{i=1}^M = ExtractPatches(C \oplus \tilde{E}, l, s), \tag{4}$$

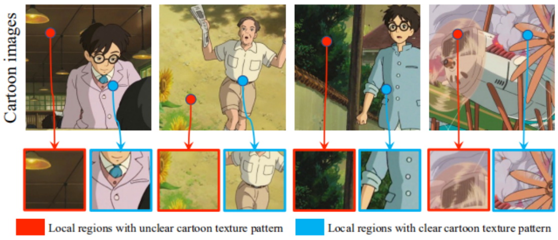


Fig. 3. The typical cartoon texture pattern manifests clearly only in partial image regions with distinct edges.

Fig. 4. Visualization of the refined edge maps $\tilde{E}$ produced during the forward pass of our CTSS module.

where $\oplus$ denotes channel-wise concat operation, l and s are respectively patch size and stride of the sliding-window-alike patch extraction process. c_p^i and e_p^i are the ith extracted patch of cartoon images C and edge maps $\tilde{E}$ respectively, they are paired and correspond to the same image location. M is the total number of the extracted patches from a mini-batch of training images, i.e., $M = N(\lfloor \frac{H-l}{s} \rfloor + 1)(\lfloor \frac{W-l}{s} \rfloor + 1)$, where H and W are height and width of training images, and N is the batch size. The extracted M image patches are sorted by the edge intensity of their paired edge patches, where the edge intensity is quantified as the pixel summation over an edge patch. After sorting, top-K image patches C_{patch} with most distinct edges are sampled:

$$t_i = \sum\nolimits_{m,n} (e_p^i)_{m,n}, i = 1, 2, ..., M, \tag{5}$$

$$C_{patch} = \{c_p^{a_j}, j = 1, ..., K | t_{a_1} \geq t_{a_2} \geq \cdots \geq t_{a_M}\}, \tag{6}$$

where m and n index pixel coordinate of each local patches, $\{a_1, a_2, ..., a_M\}$ is a permutation of $\{1, 2, ..., M\}$. The sampled top-K image patches C_{patch} contain most evident cartoon texture pattern, they serve as training data of the patch-level adversarial learning branch to promote learning cartoon texture feature. Considering that local patches can not reflect overall color distribution of images, we finally convert the sampled C_{patch} to grayscale for purpose of learning color-invariant local cartoon texture pattern. Since we use gradient based filtering method to detect edges, some local patches with no clear edges but a lot of noise can still have large accumulated gradients and thus be sampled out for patch-level training. Consequently, we apply guided filtering (Eq. 1) for edge-preserving image denoising before edge extraction (Eq. 2), this guarantees that image patches are sampled out due to clear edges instead of large noises.

It is worth mentioning that our method is more suitable than using pre-trained deep model to detect edges. Firstly, well-trained deep edge detection models tend to make high-confidence prediction to any edge pixels, the generated edge maps are less able to reflect edge intensity difference, and thus can not locate true edge-salient local regions. Secondly, our filtering based method uses only single-layer convolution to produce edge maps, which is much faster than forward propagation through pre-trained deep models.

C. Objective functions

The training of our model comprises five loss functions, they are content loss, global adversarial loss, local adversarial loss, color reconstruction loss, and total variation loss.

Content loss is used to guarantee content consistency between input photos and cartoonized results, which is realized by matching feature maps at the lth layer of the pre-trained VGG19 [46] network:

$$L_{con} = \mathbb{E}_{p_i \sim \mathcal{P}}[||VGG_l(p_i) - VGG_l(G(p_i))||_1], \tag{7}$$

where the lth layer is "conv4-4" in VGG19.

Global adversarial loss aims to capture global cartoon style through image-level adversarial learning branch. We employ LSGAN [47] loss for better stability:

$$L_{adv\_global} = L^D_{adv\_global} + L^G_{adv\_global}, \tag{8}$$

$$\begin{aligned} L^D_{adv\_global} = &\mathbb{E}_{c_i \sim \mathcal{C}}[(D_{img}(c_i) - 1)^2] + \\ &\mathbb{E}_{p_i \sim \mathcal{P}}[(D_{img}(G(p_i)))^2], \end{aligned} \tag{9}$$

$$L^G_{adv\_global} = \mathbb{E}_{p_i \sim \mathcal{P}}[(D_{img}(G(p_i)) - 1)^2]. \tag{10}$$

Local adversarial loss aims at learning local cartoon texture pattern through patch-level adversarial learning branch:

$$L_{adv\_local} = L^D_{adv\_local} + L^G_{adv\_local}, \tag{11}$$

$$\begin{aligned} L^D_{adv\_local} = &\mathbb{E}_{C_{patch}}[\frac{1}{K} \sum\nolimits_{i=1}^{K} (D_{patch}(c_p^i) - 1)^2] \\ &+ \mathbb{E}_{S_{patch}}[\frac{1}{K} \sum\nolimits_{i=1}^{K} (D_{patch}(s_p^i))^2], \end{aligned} \tag{12}$$

$$L^G_{adv\_local} = \mathbb{E}_{S_{patch}}[\frac{1}{K} \sum\nolimits_{i=1}^{K} (D_{patch}(s_p^i) - 1)^2], \tag{13}$$

where $C_{patch} = \{c_p^i\}_{i=1}^K$ are extracted top-K edge-distinct patches from $C = \{c_i\}_{i=1}^N$, $S_{patch} = \{s_p^i\}_{i=1}^K$ are top-K edge-distinct patches from $S = \{s_i\}_{i=1}^N = \{G(p_i)\}_{i=1}^N$.

Color reconstruction loss is used to retain color information after cartoonization. Following [7], we convert image from RGB to YUV format, and apply L_1 loss to Y channel and Huber Loss to U and V channels:

$$\begin{aligned} L_{col} = &\mathbb{E}_{p_i \sim \mathcal{P}}[||Y(G(p_i)) - Y(p_i)||_1 + \\ &||U(G(p_i)) - U(p_i)||_H + ||V(G(p_i)) - V(p_i)||_H], \end{aligned} \tag{14}$$

where $Y(\cdot)$, $U(\cdot)$, $V(\cdot)$ represent the three channels of an image in YUV format, and H denotes Huber Loss.

Total variation loss is used to reduce noises and artifacts of the generated results:

$$\begin{aligned} L_{tv} = &\mathbb{E}_{s_i \sim \mathcal{S}}[\frac{1}{H(W-1)} \sum_{r=1}^{H} \sum_{c=1}^{W-1} (s_{i_{r,c+1}} - s_{i_{r,c}})^2 \\ &+ \frac{1}{(H-1)W} \sum_{r=1}^{H-1} \sum_{c=1}^{W} (s_{i_{r+1,c}} - s_{i_{r,c}})^2], \end{aligned} \tag{15}$$

where H and W are height and width of generated images. The total loss function can be decomposed into a generator part and a discriminator part:

$$L_{gen} = \lambda_{global} L^{G}_{adv\_global} + \lambda_{local} L^{G}_{adv\_local} + \lambda_{con} L_{con} + \lambda_{col} L_{col} + \lambda_{tv} L_{tv}, \quad (16)$$

$$L_{dis} = \lambda_{global} L^{D}_{adv\_global} + \lambda_{local} L^{D}_{adv\_local}, \quad (17)$$

where L_{gen} is minimized to optimize the generator G, L_{dis} is minimized to jointly optimize the two discriminators D_{img} and D_{patch}. L_{gen} and L_{dis} are minimized alternately to form the adversarial training framework.

自己看英文描述还是可以很容易的理解本文的；

由于独特的卡通纹理模式在边缘不同的局部区域视觉上更突出，我们的CTSS模块自适应地从每个输入的小批图像中采样具有最不同边缘的局部图像块，实现细节如图2所示

对于边缘处理的步骤原理：考虑到局部斑块不能反映图像的整体颜色分布，我们最终将采样的Cpatch转换为灰度，以学习颜色不变的局部卡通纹理模式。由于我们使用基于梯度的滤波方法来检测边缘，一些没有清晰边缘但有大量噪声的局部斑块仍然可以有较大的累积梯度，因此被采样进行斑块级训练。因此，我们应用引导滤波(Eq。1)在边缘提取前的边缘保留图像去噪。2)，这就保证了图像补丁是由于清晰的边缘而被采样出来的，而不是由于较大的噪声。

tips：本模型可以考虑用于物品检测的边缘检测吗。请自己留意

内容损失基本上成熟了使用vgg19进行pre训练就行。

gan损失采用lsgan（最小二乘法gan）的损失函数

Color reconstruction loss is used to retain color information after cartoonization. Following [7], we convert image from RGB to YUV format, and apply L_1 loss to Y channel and Huber Loss to U and V channels:

$$L_{col} = \mathbb{E}_{p_i \sim \mathcal{P}}[||Y(G(p_i)) - Y(p_i)||_1 + ||U(G(p_i)) - U(p_i)||_H + ||V(G(p_i)) - V(p_i)||_H], \quad (14)$$

where $Y(\cdot)$, $U(\cdot)$, $V(\cdot)$ represent the three channels of an image in YUV format, and H denotes Huber Loss.

如图所示，YUVformat及其转换操作。[什么是YUV?]

> 引用标记

part and a discriminator part:

$$L_{gen} = \lambda_{global} L^{G}_{adv\_global} + \lambda_{local} L^{G}_{adv\_local} + \lambda_{con} L_{con} + \lambda_{col} L_{col} + \lambda_{tv} L_{tv}, \tag{16}$$

$$L_{dis} = \lambda_{global} L^{D}_{adv\_global} + \lambda_{local} L^{D}_{adv\_local}, \tag{17}$$

where L_{gen} is minimized to optimize the generator G, L_{dis} is minimized to jointly optimize the two discriminators D_{img} and D_{patch}. L_{gen} and L_{dis} are minimized alternately to form the adversarial training framework.

然后是正常的gan优化流程

conclusion：本文提出了一种用于图像制图的深度生成模型。我们用补丁级学习分支补充了正常的图像级对抗训练，并仅在卡通纹理存在的局部区域自适应地约束补丁级对抗学习，以增强卡通纹理模式的捕获。为此，提出了一种新的卡通纹理显著采样器（CTSS）模块，以动态采样包含最显著卡通纹理特征的局部图像块。通过将这种纹理显著性的自适应注意纳入对抗性学习，我们的方法能够明显地转移更抽象和生动的卡通风格。

2. 学习了图像的基本知识：灰度 rgb yuv等表示方法；图像的高频和低频特征等：

一、图像高频信号和低频信号的理解

1.1 图像中的低频信号和高频信号也叫做低频分量和高频分量。简单一点说，图像中的高频分量，指的是图像强度（亮度/灰度）变化剧烈的地方，也就是我们常说的边缘（轮廓）；图像中的低频分量，指的是图像强度（亮度/灰度）变换平缓的地方，也就是大片色块的地方。人眼对图像中的高频信号更为敏感。 图像的高低频是对图像各个位置之间强度变化的一种度量方法。低频分量:主要对整副图像的强度的综合度量. 高频分量:主要是对图像边缘和轮廓的度量。如果一副图像的各个位置的强度大小相等,则图像只存在低频分量,从图像的频谱图上看,只有一个主峰,且位于频率为零的位置。 如果一副图像的各个位置的强度变化剧烈,则图像不仅存在低频分量,同时也存在多种高频分量,从图像的频谱上看,不仅有一个主峰,同时也存在多个旁峰。

1.2直观认识 假设在正弦波中有一个毛刺，并且正弦波的变化非常的缓慢，频率较低，在正弦波上有一个毛刺，这个毛刺在短时间内就完成了一个变化周期，频率较高。所以我们就把这里的正弦波称为低频信号，而毛刺就称为高频信号。如果要对这个曲线平滑滤波的话，效果就是把毛刺滤掉，也就是说，平滑滤波的操作会将高频信号去除而低频信号保留，也就是我们常说的低通滤波器了。 最简单的低通滤波器的实现就是中值或者均值滤波器。 由以上的认识推广到二维图像上，也就不难知道为什么会将图像上变化剧烈的地方叫做高频信号，而变化平缓的地方叫做低频信号了。

二、图像频率的理解

1. 不同频率信息在图像结构中有不同的作用。图像的主要成分是低频信息，它形成了图像的基本灰度等级，对图像结构的决定作用较小；中频信息决定了图像的基本结构，形成了图像的主要边缘结构；高频信息形成了图像的边缘和细节，是在中频信息上对图像内容的进一步强化。

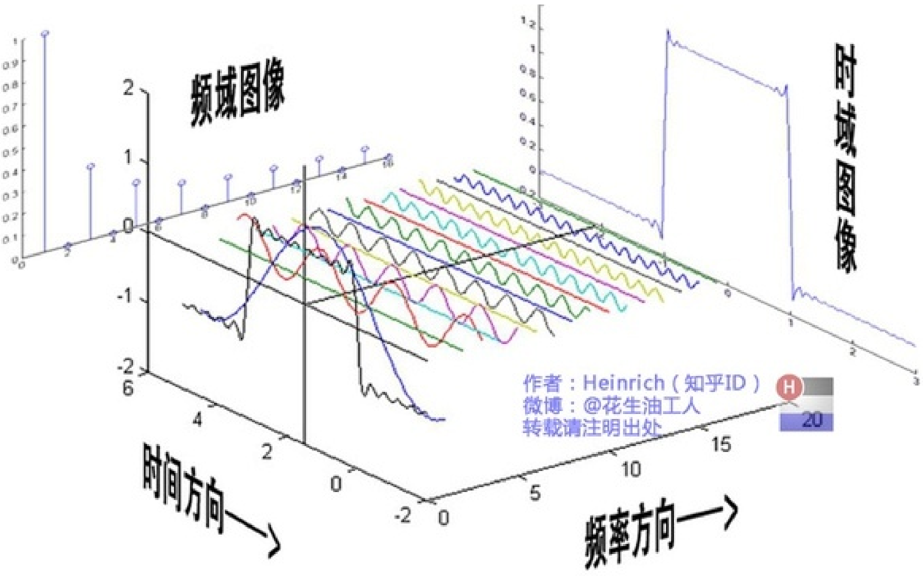
2. 图像的频率是表征图像中灰度变化剧烈程度的指标，是灰度在平面空间上的梯度。如：大面积的沙漠在图像中是一片灰度变化缓慢的区域，对应的频率值很低；而对于地表属性变换剧烈的边缘区域在图像中是一片灰度变化剧烈的区域，对应的频率值较高。

3. 对图像而言，图像的边缘部分是突变部分，变化较快，因此反应在频域上是高频分量；图像的噪声大部分情况下是高频部分；图像平缓变化部分则为低频分量。也就是说，傅立叶变换提供另外一个角度来观察图像，可以将图像从灰度分布转化到频率分布上来观察图像的特征。

4. 图像进行二维傅立叶变换得到频谱图，就是图像梯度的分布图,当然频谱图上的各点与图像上各点并不存在一一对应的关系，即使在不移频的情况下也是没有。傅立叶频谱图上我们看到的明暗不一的亮点，实际是上图像上某一点与邻域点差异的强弱，即梯度的大小，也即该点的频率的大小（可以这么理解，图像中的低频部分指低梯度的点，高频部分相反）。

5. 图像的频率，不是图像上某一个点的频率，它反映了反应了图像像素变化的快慢，也就是说，在某一区域变化的非常大非常的快，那这一区域就携带有一定的高频的信息。图像的高频信息越多，图像的细节特征也就越多。

3. 重修傅里叶变换



4. 学了一点lua