

PSNR 的实现过程

PSNR 其实仅仅根据所给的公式，带入计算即可，基本没有其他特殊问题

SSIM 的实现过程

根据论文所给的公式，为了更好地计算 SSIM，我们这里采用加权平均数地方式来进行处理，即对于每一个窗口区间，分别求出这个窗口的 SSIM 值，利用高斯核赋予权值，本质上就是一个滤波的过程，在对每个窗口求出对应的 SSIM 值后，再整体相加取均值求得整张图片的 SSIM 值

双三次插值简要思路

开始的预处理阶段类似于双三次插值，即左上角对齐之后，将目标的像素点映射到原图的某一具体位置，此时可以通过

$$p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 p(x_i, y_j) W(x - x_i) W(y - y_j),$$

来对这个像素点进行赋值，通过取周围 16 个点，并取加权和，就可以得知目标的像素点的值，之后再进行直方图均衡化，便可以获得想要的结果。

超分辨率算法实现思路

其实所有的思路都是从作者的论文中获得的，这里我将分步骤将我所实现的流程描述出来：

1. 将 HR 图像转成 LR 图像

这里的实现很简单，即利用论文的公式

$$I_l = (I_h \otimes G) \downarrow_s,$$

将图片从高分辨率的图像经过滤波和下采样后, 取得低分辨率的图片

2. 取得 LR patch

从每一张图片中, 我们都可以获得很多个 LR patch, 它们是 7×7 的像素块, 即从最左上角的像素块出发, 每次往后移动一个像素, 从而获得无数多个 LR patch

3. 获得 feature 和中心点

每一个 LR patch 去掉周围的 4 个点后, 再减去均值, 便可以得到对应的 feature, 这个 feature 可以看作是 patch 的特征, 之后会在聚类中使用到; 而为了方便以后的使用, 我把每一个 patch 的中心点保存下来, 方便之后的使用, 即方便之后寻找 LR 和 HR 的映射

4. 聚类

考虑到电脑性能问题, 这里选取第 3 步中获得的一部分样本进行聚类处理, 并根据样本数目对每个类的中心点进行排序存储

5. 获取映射函数

为了求得 LR 和 HR patch 之间的映射, 这里对所有样本, 取得对应地 LR 和 HR 的映射, 并根据

$$C^* = \underset{C}{\operatorname{argmin}} \left\| W - C \begin{pmatrix} V \\ 1 \end{pmatrix} \right\|^2,$$

求得回归系数，从而为之后的测试提供数据

6. 测试数据

对于每一个图片，我们同样对图片进行 patch 划分，并将这里的每一个 patch 归类，根据类里面的回归系数，求出相应的高分辨率图像，由于高分辨率图像每个像素点可能会被重复计算，因此这里我们取均值进行还原。从论文中看，由于人眼对亮度的变化会对色彩更加敏感，因此我们对图像的亮度通道进行运算，其他的按照双三次插值求得对应 size 的图片即可

利用双三次插值得出的量化指标

	PSNR(bicubic)	SSIM	PSNR(paper)	SSIM
Baboon	21.6768	0.5208		
Barbara	24.7461	0.7431		
Bridge	23.2377	0.6587		
Coastguard	24.4173	0.6027		
Comic	21.9525	0.7156		
Face	28.0589	0.7427		
Flowers	26.5531	0.7917		
Foreman	27.6341	0.8845		
Lenna	29.3160	0.8309		

Man	25.8048	0.7246		
Monarch	28.5066	0.9078		
Pepper	28.3664	0.8301		
Ppt3	22.1350	0.8763		
Zebra	25.2752	0.7904		
average	25.5628	0.7586		

结果分析

再 bicubic 的结果中，我发现结果与原图的差别很大，同时，越复杂的地方效果越差，而平滑一点的地方效果会较好。这与该算法的过程也相关。

而在论文提到的算法中，当取得的 cluster 数目越多，结果也会越好，细节部分也更好，由于我这边代码还没有完全优化完成，因此我的图片还有问题，也没有计算评估值，此时的评估值一定不准确的。但是从结果看来，效果较 bicubic 来说，还是更好一些的。同时，我也尝试了不同的 cluster 数目来计算，结果也十分显然，数目越多，还原度就越高，因为较多的数目会带来更多的分类，因此 patch 的所属类别也就越精细。

改进

从自然景物来看，一个景物一定是连续的，因此一个点与周围的价值一定是相关的，所以插值算法可以考虑周围更多的点来还原，而论文中的算法可以考虑对类别进行区分，比如动

漫图片的还原一定不能使用自然景物的聚类来计算，效果一定是不好的，所以除了用更多的类别来计算，对图片类别的区分和利用也是较为重要的一方面。