

# Final Project: 图像超分辨率

数字图像处理课程助教， 中山大学

**欢**迎进入数字图像处理大作业！数字图像处理课程即将结束，我们希望你能享受这个难忘的经历并且从中学到有用的知识。最后，我们准备深入了解并开发一个有趣的应用——图像超分辨率。这次大作业与中中考一样，占总评相当大的比例，所以请认真对待。

## 1 引言

通过之前的作业，你已经意识到，当你直接放大一张低分辨率图像的时候，你将得到一张模糊的图像。以往的你，为了看清楚低分辨率图像的细节，例如下图蝴蝶的翅膀，你往往会点击放大按钮把图像放大，然而，放大之后模糊的细节不尽人意。幸好，你参加了数字图像处理课程，现在的你能够开发出图像超分辨率算法，可以从低分辨率图像生成出一张高分辨率图像。

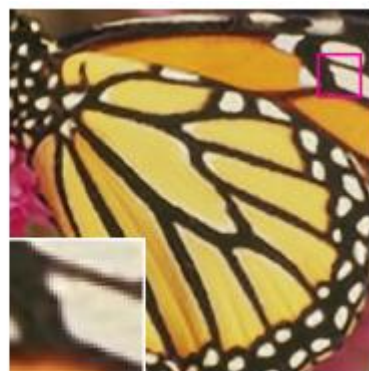
针对这次大作业，我们将在附件中提供一个简单而有效的图像超分辨率方法，作为基本任务。另外我们也提供难度更高的方法，作为附加任务。这次大作业的成绩将很大程度上取决于最终效果。



(a) 高分辨率图像



(b) 双三次插值放大



(c) 超分辨率方法放大

图 1. 蝴蝶翅膀放大 3 倍，其中(a)是高分辨率图像，(b)和(c)是低分辨率图像

## 2 要求

请解压附件中的 Train 和 Set14 文件，其中 Train 的图像用于超分辨率算法的训练与参数调整，而 Set14 的图像用于测试超分辨率算法的效果。

### 2.1 基本任务（100 分）

我们常用**峰值信噪比 PSNR**(Peak Signal-to-Noise Ratio)和**结构相似性指标 SSIM**(Structural Similarity Index)这两个指标来反映两张尺寸相同的图像的相似程度。PSNR 值和 SSIM 值越大，意味着这两张图片越相似。如图 2 所示，图 2(c)的 PSNR 值和 SSIM 值都比图 2(b)的大，而我们通过主观判断也可以看出，图 2(c)比图 2(b)跟原图更相似。



(a)原图 (PSNR, SSIM)

(b) (27.08, 0.7508)

(c) (27.32, 0.7606)

图 2. (a)是高分辨率原图，(b)和(c)是低分辨率图像通过不同的超分辨率算法得到的近似图

在图像超分辨率任务中，我们将用这两个指标作为评价指标。如果原图和经过超分辨率后得到的高分辨率图像的 PSNR 值和 SSIM 值越大，则证明这种超分辨率方法越好。在接下来的任务 1 和任务 2 中，我们主要完成这两个评价指标的实现。

1. （10 分）实现超分辨率评价指标——峰值信噪比 PSNR(Peak Signal-to-Noise Ratio)，函数格式是 "PSNR(input\_img1, input\_img2) → output"，该函数返回图像 input\_img1 和图像 input\_img2 的 PSNR 值。给定尺寸  $M \times N$  的原图  $X$  及其噪声近似图像  $Y$ ，则它们的均方差 MSE(Mean Squared Error)可以表示成：

$$MSE(X,Y) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [X(i,j) - Y(i,j)]^2$$

它们的  $PSNR$  值定义为:

$$PSNR(X,Y) = 20 \cdot \log_{10}(\frac{MAX_I}{\sqrt{MSE(X,Y)}})$$

其中  $MAX_I$  是图像的灰度级别 - 1 (对于 8-bit 灰度图, 若灰度值区间为[0,255]则取值 255).

提示: 对于 RGB 彩色图像, 我们可以先将图像由 RGB 色彩空间转换到 YCbCr 色彩空间, 然后只用 Y 通道来计算  $PSNR$ 。Matlab 示例代码为:

```
img = rgb2ycbcr(img);
img = img(:, :, 1);
```

2. (10 分) 实现超分辨率的另一个评价指标——结构相似性指标  $SSIM$  (Structural Similarity Index), 函数格式是 " $SSIM$  (input\_img1, input\_img2)  $\rightarrow$  output", 该函数返回图像 input\_img1 和图像 input\_img2 的  $SSIM$  值。给定尺寸  $M \times N$  的原图  $X$  及其噪声近似图像  $Y$ , 他们的  $SSIM$  可以定义为:

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}$$

- $\mu_X$  是图像  $X$  的平均值
- $\mu_Y$  是图像  $Y$  的平均值
- $\sigma_X^2$  是图像  $X$  的方差
- $\sigma_Y^2$  是图像  $Y$  的方差
- $\sigma_{XY}$  是  $X$  和  $Y$  的协方差
- $c_1 = (k_1 L)^2$ ,  $c_2 = (k_2 L)^2$  是两个用来维持稳定的常数
- $L$  是图像的灰度级别 - 1 (对于 8-bit 灰度图, 若灰度值区间为[0,255]则取值 255)
- $k_1$  和  $k_2$  默认分别取 0.01 和 0.03.

提示:  $SSIM$  的详细计算方式请参照 <https://ece.uwaterloo.ca/~z70wang/research/ssim/>, 对于 RGB 彩色图像处理方法同上。请在报告中写明  $SSIM$  的实现思路。

3. (20 分) 实现基于双三次插值(bicubic interpolation)的图像缩放算法。函数格式是 "bicubic(input\_img, height, width) → output\_img", 该函数返回双三次插值后的结果图像。请在报告中描述双三次插值算法(bicubic interpolation), 并简要写出实现思路。

提示: 通过第一次作业, 大家已经了解到如何利用双线性插值(bilinear interpolation)进行图像缩放。现在开始实现一种更加复杂插值方式——双三次插值, 该算法能够产生比双线性插值更清晰的图像。双三次插值利用了邻近的 16 个点来估计给定位置的灰度值。令  $(x, y)$  是我们想要赋以灰度值的原图位置坐标, 并令  $p(x, y)$  表示其灰度值, 那么利用双三次插值计算  $p(x, y)$  的公式如下:

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

其中  $p(x_i, y_j)$  是点  $(x, y)$  的邻近点灰度值,  $W(x)$  是邻近点对应的权重, 它按照下面公式计算:

$$W(x) = \begin{cases} 1.5|x|^3 - 2.5|x|^2 + 1 & 0 \leq |x| \leq 1 \\ -0.5|x|^3 + 2.5|x|^2 - 4|x| + 2 & 1 < |x| \leq 2 \\ 0 & otherwise \end{cases}$$

关于双三次插值的详细解释请参考附件中的 "Cubic convolution interpolation for digital image processing" 以及 "ImageZooming".

4. (10 分) 用基于双三次插值的图像超分辨率算法进行实验。具体实验步骤如下:
- a) 对 Set14 的每一张图像  $I_{HR}$ , 利用双三次插值算法, 把原图缩小至原来尺寸的  $1/3$  得到图像  $I_{LR}$
  - b) 再次利用双三次插值算法将  $I_{LR}$  上采样至  $I_{LR}$  尺寸的三倍得到图像  $I_{BI}$  (与  $I_{HR}$  尺寸相同)
  - c) 计算  $PSNR(I_{HR}, I_{BI})$  和  $SSIM(I_{HR}, I_{BI})$ , 并填写在表格 1 中。请对照所得到的结果是否与论文 3(ECCV 2014)的结果相近。
5. (50 分) 仔细阅读并实现论文 1 (ICCV 2013) 的超分辨率算法。仿照第 4 问, 对 Set14 每一张低分辨率图像  $I_{LR}$  (原图  $I_{HR}$  经过双三次插值缩至原尺寸  $1/3$  所得的图像), 通过超分辨率算法获得图像  $I_{SR}$  ( $I_{SR}$  的尺寸和原图  $I_{HR}$  尺寸一样), 并计算出  $PSNR(I_{HR}, I_{SR})$  和  $SSIM(I_{HR}, I_{SR})$ , 填写在表格 1 中。同时也请将每张图像进行超分辨运算的时间填写在表格 1 中。请在报告中描述算法内容, 并简要写出实现思路。

提示: 论文 1 的超分辨率方法涉及聚类, 聚类的介绍可参考附件 Clustering.pptx。聚类可以使用函数库, 例如 matlab 的聚类函数库使用方法可以参考 <https://cn.mathworks.com/help/stats/kmeans.html>

## 2.2 高级任务（可选）（30 分）

1. 选择论文 2, 3 或 4 的其中一篇，也可以自行寻找适当方法（需注明出处），并将其实现。仿照 2.1 的第 5 问，计算出相应的  $PSNR$  和  $SSIM$ ，同时记录相应的运行时间。把这些结果也填写在表格 1 中。请在报告中描述算法内容，并简要写出实现思路。
2. 分析当前实现的算法优缺点，在效果与效率方面跟 2.1.5 实现的超分辨率方法进行对比，并解释原因。

表格 1：不同算法在 Set14 测试集上的  $PSNR$ ,  $SSIM$  以及运行时间

Set14 images	双三次插值算法		基本任务超分辨率算法			高级任务超分辨率算法		
	$PSNR$	$SSIM$	$PSNR$	$SSIM$	$Time$	$PSNR$	$SSIM$	$Time$
baboon								
barbara								
bridge								
coastguard								
comic								
face								
flowers								
foreman								
lenna								
man								
monarch								
pepper								
ppt3								
zebra								
average								

## 2.3 编程语言

你可以选择任意的编程语言。如果你选择了深度学习相关的方法，例如论文 3 和 4，那么你不允许使用 caffe 与 matconvnet。

## 2.4 提交

1. 完整的代码。

2. 实验报告(PDF 格式), 包括 1) 简要介绍你所实现的方法, 2) 各个算法的实现过程, 3) 实现结果与分析 4) 分析你所实现的算法的缺点, 并提出你可能的改进方案。
3. 描述算法实现思路的时候, 不要简单地复制粘贴代码。
4. 作业文件夹命名格式为: 编程语言\_学号\_名字\_大作业, 以 zip 压缩包提交。