

基于 SRResNet 与 SRGAN 的图像超分辨率重建算法研究与应用

20121802 严昕宇

摘要：图像超分辨率技术，是指从低分辨率图像重建得到相应的高分辨率图像。目前的传统图像超分辨率技术在恢复低分辨率图像时存在纹理模糊，物体轮廓不明显的现象。因此，本文对深度学习中的 SRResNet 和 SRGAN 模型进行分析，基于 Pytorch 框架完成 SRGAN 算法的复现，并与传统算法进行对比。

关键词：图像重建；超分辨率；SRResNet；SRGAN；深度学习

1. 项目背景和意义

图像分辨率是一组用于评估图像中蕴含细节信息丰富程度的性能参数。相较于低分辨率图像，高分辨率图像通常包含更大的像素密度、更丰富的纹理细节及更高的可信赖度。要获得高分辨率图像，最直接的办法是采用高分辨率图像传感器，但由于传感器和光学器件制造工艺和成本的限制，在很多场合和大规模部署中很难实现。

图像的超分辨率(Super Resolution, SR)重建技术指的是利用数字图像处理、计算机视觉等领域的相关知识，将给定的低分辨率(Low Resolution, LR)图像，通过特定的算法恢复成相应的高分辨率图像(High resolution, HR)。因此，通过超分辨率技术获取高分辨率图像，具有重要的现实意义。

本文以 *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network* 这篇论文为主线，依次分析其中提出的 SRResNet 和 SRGAN 两种图像超分辨率重建算法的实现原理，同时基于 Pytorch 深度学习框架完成 SRGAN 算法的复现，并与传统超分辨率重建算法进行对比。

2. 传统超分辨率重建算法

传统的超分辨率重建算法主要依靠基本的数字图像处理技术进行重建，常见的有基于插值、退化模型、学习的超分辨率重建。其中，基于插值的方法将图像上每个像素都看做是图像平面上的一个点，那么对超分辨率图像的估计则是利用已知的像素信息为平面上未知的像素信息进行拟合的过程，这通常由一个预定义的变换函数或者插值核来完成。常见的基于插值的方法包括最近邻插值法、双线性插值法和双三线性插值法等。

以最近邻插值法为例，只需要通过映射，将原始图片中的像素值映射到放大(或者缩小)后的图片中的每一个位置上即可，而不需要通过计算来得到放大后图片中的每一个像素值。

像素值映射的公式如下所示：

$$\begin{aligned}src_x &= dst_x / scale \\src_y &= dst_y / scale\end{aligned}$$

其中： src_x 、 src_y 表示原始图像中的坐标， dst_x 、 dst_y 表示放大图片中的坐标， $scale$ 表示放缩倍数。当然，在计算过程中会出现小数的情况，这时将小数四舍五入即可。

以下使用代码实现了基于插值的超分辨率重建，效果如图 2.1 所示。可以发现基于插值的方法计算存在着一些明显的缺陷。首先，它假设像素灰度值的变化是一个连续的、平滑的过程，但实际上这种假设并不完全成立。其次，在重建过程中，仅根据一个事先定义的转换函数来计算超分辨率图像，不考虑图像的降质退化模型，往往会导致复原出的图像仍有模糊、锯齿等现象，因而传统超分辨率重建算法的应用场景较为局限。

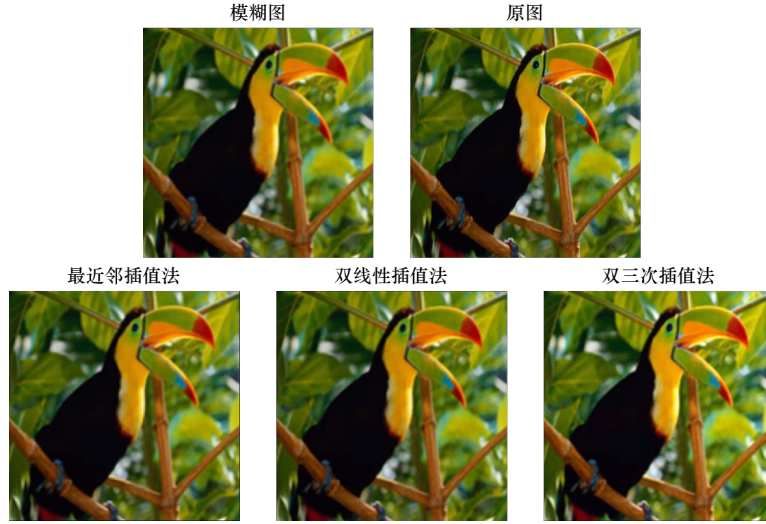


图 2.1 基于插值的超分辨率重建效果图

3. SRResNet 算法原理

3.1 SRResNet 结构

SRResNet 使用深度残差网络来构建超分重建模型，主要包含两部分：深度残差模块、子像素卷积模块。深度残差模块用来进行高效的特征提取，可以在一定程度上削弱图像噪点。子像素卷积模块主要用来放大图像尺寸。完整的 SRResNet 网络结果如下图所示：

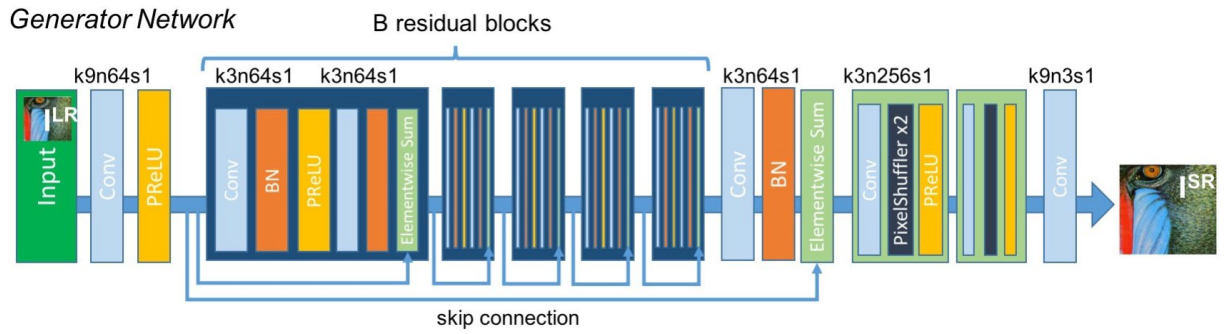


图 3.1 SRResNet 网络结构

上图中， k 表示卷积核大小， n 表示输出通道数， s 表示步长。除了深度残差模块和子像素卷积模块以外，在整个模型输入和输出部分均添加了一个卷积模块用于数据调整和增强。

需要注意的是，SRResNet 模型使用 MSE 作为目标函数，也就是通过模型还原出来的高分辨率图像与原始高分辨率图像的均方误差，公式如下：

$$L = \frac{1}{n} \sum_{i=1}^n \|F(X_i; \Theta) - Y_i\|^2$$

3.2 深度残差模块

最早的采用深度学习进行超分重建的算法是 SRCNN 算法，其原理很简单，对于输入的一张低分辨率图像，首先使用双立方插值将其放大至目标尺寸，然后利用一个三层的卷积神经网络去拟合低分辨率图像与高分辨率图像之间的非线性映射，最后将网络输出的结果作为重建后的高分辨率图像。在 SRCNN 之后，有不少研究人员尝试加深网络结构以期取得更佳的重建性能，但是越深的模型越无法很好地收敛。

此问题由 2015 年何凯明团队提出的深度残差网络得以有效解决。深度残差网络(Deep Residual Network, ResNet)在传统卷积神经网络中加入了残差学习(Residual Learning)，解决了深层网络中梯度

弥散和精度下降(训练集)的问题，使网络能够越来越深，既保证了精度，又控制了速度。

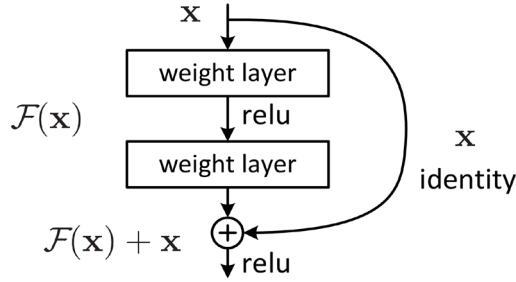


图 3.2 残差网络原理图

上图为残差网络的原理图，可以看到一根线直接跨越两层网络(跳链)，将原始数据 x 带入了输出中，此时 $F(x)$ 预测的是一个差值。基于残差学习的网络结构，就可以按照 SRCNN 的思路构建用于超分重建的深度神经网络。

SRResNet 采用了 16 个深度残差模块进行图像的特征抽取(即图 3.1 中的 B residual blocks 部分)，多次运用跳链技术将输入连接到网络输出。由于采用了深度模型，相比浅层模型能够更有效的挖掘图像特征，在性能上可以超越浅层模型算法。但值得注意的是，上述模型每层仅仅改变了图像的通道数，并没有改变图像的尺寸大小，从这个意义上来说这个网络可以认为是 SRCNN 的修复模型。下面会介绍如何在此基础上再增加一个子模块用来放大图像，从而构建一个完整的超分重建模型。

3.3 子像素卷积模块

子像素卷积(Sub-pixel Convolution)，又称为 Pixel Shuffle(像素清洗)是一种巧妙的图像及特征图放大方法。图 3.3 很直观得表达了子像素卷积的流程。假设有对原图放大 3 倍，那么需要生成 $3^2=9$ 个同等大小的特征图，即通道数扩充了 9 倍。然后将九个同等大小的特征图拼成一个放大 3 倍的大图，这就是子像素卷积操作了。

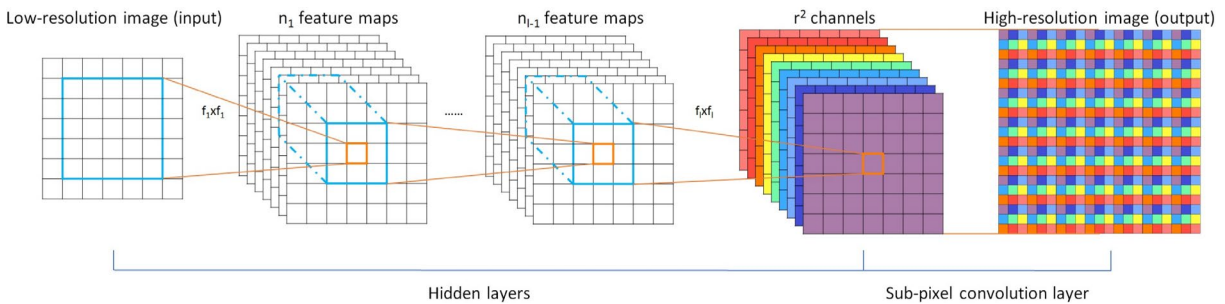


图 3.3 子像素卷积示意图

通过像素清洗，特征的通道数重新恢复为原来输入时的大小，但是每个特征图的尺寸变大了。这里注意到每个像素的扩展方式由对应的卷积来决定，此时卷积的参数是需要学习的，因此，相比于手工设计的放大方式，这种基于学习的放大方式能够更好的去拟合像素之间的关系。SRResNet 也利用子像素卷积来放大图像。具体的，SRResNet 网络中添加了两个子像素卷积模块，每个子像素卷积模块使得输入图像放大 2 倍，因此这个模型最终可以将图像放大 4 倍。

4. SRGAN 算法原理与实现

SRResNet 算法是一个单模型算法，从图像输入到图像输出中间通过各个卷积模块的操作完成，整个结构比较清晰。但其所采用的 MSE 会使得超分重建出的图像过于平滑，丢失了细节纹理信息。而 SRGAN 模型则可以在纹理细节丢失的情况下，“无中生有”的重建出这些信息。

4.1 SRGAN 结构

SRGAN 分为两部分：生成器模型(Generator)和判别器模型(Discriminator)。生成器模型采用了 SRResNet 完全一样的结构(如图 3.1),只是在计算损失函数时需要利用截断的 VGG19 模型进行计算。

判别器模型对原始高清图像或者重建的高清图像进行判断,判断图像到底是不是生成器创造出来。本质上是一个分类模型,因此判别器的最终输出是一个 1 维的张量。判别器模型中间部分使用了多个卷积模块进行特征提取。判别器模块结构如下所示:

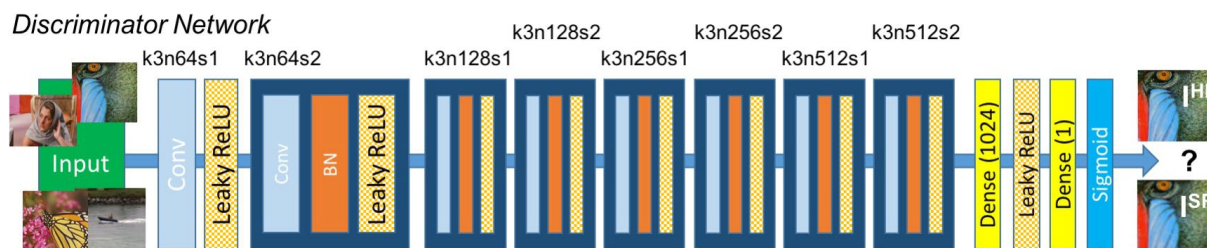


图 4.1 SRGAN 判别器结构

4.2 生成对抗网络(GAN)

GAN 的主要灵感来源于博弈论中博弈的思想,应用到深度学习上来说,就是构造两个深度学习模型:生成网络 G(Generator)和判别网络 D(Discriminator),然后两个模型不断博弈,进而使 G 生成逼真的图像,而 D 具有非常强的判断图像真伪的能力。生成网络和判别网络的主要功能是:

- G 是一个生成式的网络,它通过某种特定的网络结构以及目标函数来生成图像;
- D 是一个判别网络,判别一张图片是不是“真实的”,即判断输入的照片是不是由 G 生成;

4.3 感知损失

为了防止重建图像过度平滑,SRGAN 重新定义了损失函数,并将其命名为感知损失(Perceptual loss)。感知损失有两部分构成:

$$\text{感知损失} = \text{内容损失} + \text{对抗损失}$$

对抗损失就是重建出来的图片被判别器正确判断的损失,这部分内容跟一般的 GAN 定义相同。SRGAN 的一大创新点就是提出了内容损失。SRGAN 希望让整个网络在学习的过程中更加关注重建图片和原始图片的语义特征差异,而不是逐个像素之间的颜色亮度差异。以往我们在计算超分重建差异时使用的是 MSE 准则,SRGAN 算法提出者认为这种方式只会过度的让模型去学习这些像素差异,而忽略了重建图像的固有特征。实际的差异计算应该在图像的固有特征上计算。

但是这种固有特征怎么表示呢?其实很简单,已经有很多模型专门提出来提取图像固有特征然后进行分类等任务。我们只需要把这些模型中的特征提取模块截取出来,然后去计算重建图像和原始图像的特征,这些特征就是语义特征了,然后再在特征层上进行两幅图像的 MSE 计算。在众多模型中,SRGAN 选用了 VGG19 模型,其截取的模型命名为 truncated_vgg19。

4.4 Pytorch 实现

(a) 实验环境

本实验使用 Ubuntu 操作系统,基于 PyTorch 框架,使用 RTX 2080 Ti 显卡进行训练。其中,参数的设置为: Batch Size = 256, Epoch = 100

SoundLab		Sun May 14 14:26:33 2023		470.182.03	
[0]	NVIDIA GeForce RTX 2080 Ti	44°C, 97 %	10109 / 11019 MB	jj(9949M) gdm(18M) gdm(17M) lfy(85M) lfy(15M) lfy(9M)	
[1]	NVIDIA GeForce RTX 2080 Ti	32°C, 0 %	13 / 11019 MB	gdm(4M) lfy(4M)	
[2]	NVIDIA GeForce RTX 2080 Ti	31°C, 0 %	13 / 11019 MB	gdm(4M) lfy(4M)	
[3]	NVIDIA GeForce RTX 2080 Ti	31°C, 0 %	13 / 11019 MB	gdm(4M) lfy(4M)	
[4]	NVIDIA GeForce RTX 2080 Ti	31°C, 0 %	13 / 11019 MB	gdm(4M) lfy(4M)	
[5]	NVIDIA GeForce RTX 2080 Ti	30°C, 0 %	13 / 11019 MB	gdm(4M) lfy(4M)	
[6]	NVIDIA GeForce RTX 2080 Ti	29°C, 0 %	13 / 11019 MB	gdm(4M) lfy(4M)	
[7]	NVIDIA GeForce RTX 2080 Ti	30°C, 0 %	13 / 11019 MB	gdm(4M) lfy(4M)	

图 4.2 实验环境

(b) 训练结果

图 4.3 分别展示了整个训练过程中生成损失、内容损失和判别损失的变化曲线。

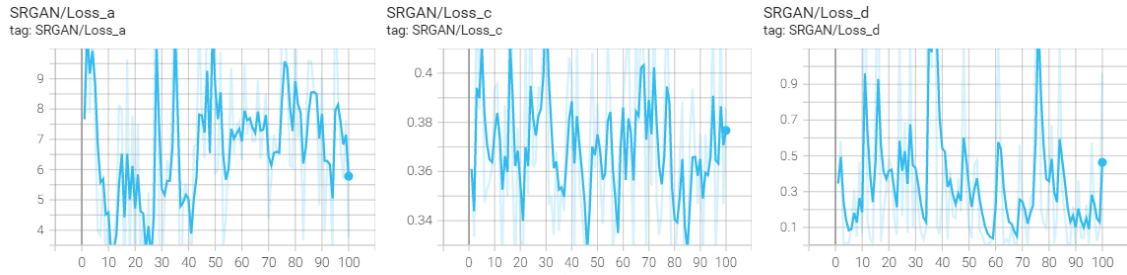


图 4.3 变化曲线

从上图中可以看到，SRGAN 的收敛曲线非常不平稳，无法通过查看 Loss 来说明 GAN 训练得如何。但可从每次 Epoch 训练样本重建效果上进行查看。如图 4.4 是 11 Epoch 时的效果，其中第一部分为低分辨率图，第二部分为超分重建图，第三部分为原始高清图，可以观察到重建效果较好。



图 4.4 11 Epoch 时训练样本的重建效果

(c) 评估

表 4.1 测试结果

	Set5	Set14	BSD100
PSNR(s)	29.057	25.426	24.248
SSIM(s)	0.839	0.688	0.628
单张图片平均用时(s)	0.103	0.070	0.039

上表中结果与论文中的值较为接近。可以看到，其 PSNR 和 SSIM 效果并不好，这是因为 SRGAN 本质上就不是为了 PSNR 和 SSIM 指标而设计优化的。为定量评价，论文中新设计了 MOS 指标，即让多位观察者采用主观评价的方式对重建效果图进行打分，最后将分数作平均并以此作为评价指标。

(d) 测试



图 4.5 重建效果

5. 总结

本文详细讲述了超分重建的概念，针对有代表性的 SRResNet 和 SRGAN 算法，分别进行了原理剖析，并基于 Pytorch 实现。从最终的效果上来看，基本可以达到原论文中的效果。

参考文献

- [1] Ledig C, Theis L, F Huszár, et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network[C]// IEEE Conference on Computer Vision and Pattern Recognition. 2017.