哈尔滨工业大学

**<<模式识别与深度学习>>**

**实验6 实验报告**

**(2020春季学期)**

|  |  |
| --- | --- |
| **成员1：** | **1170300321 欧龙燊** |
| **成员2：** | **1170300319 蒙栩** |
| **成员3：** |  |

# 实验目标

我们小组打算做一个动漫风格桌面壁纸高清大图生成器。

该程序能够输入低分辨率的、或者经过有损压缩的动漫风格图片，输出高分辨率的、符合动漫风格要求的图片。

## 选题起因

日常生活中，对动漫圈比较感兴趣的同学们有时会想要使用自己喜欢的动漫人物作为电脑桌面壁纸。

然而高清大图并不容易获得，因为并不是所有动漫爱好者都愿意花钱购买画手的作品。我们从图片搜索引擎得到的大多数图源都是“二手”的，曾经被降低了分辨率，或者经过了有损压缩。

随着液晶显示器技术发展，我们的个人电脑屏幕分辨率也逐渐在增加，dpi不断提升，物理尺寸也在不断变大。因此，我们对桌面壁纸的清晰度要求也在随之不断提升。

如果能直接从易得的低分辨率图片来生成具有更多细节的高分辨率壁纸，比起花时间寻找高清图源和花钱购买原图，是一个更加诱人的选择。本小组组员之一一直在使用一个动漫风格超分辨率系统BigJPG。该系统作者近日关闭了免费使用通道。为弥补这一遗憾，我们打算构造一个具有类似功能的系统。

## 思路

想要做好动漫风格图片处理，首先要了解动漫图像风格。想象一下一张漫画是如何被创作的——先勾线、后上色。大多数动漫图像由边缘线条和色块构成，是一种介于“现实世界照片”和“简单几何图形”之间的风格。一个优良的动漫风格超分辨率系统的输出应该具备的特点是：线条光滑、几何形状棱角锐利、色块区域平滑无斑点。

传统插值拉伸放大几乎都无法避免锯齿、线条模糊、色块、马赛克等问题。所以我们选择使用卷积神经网络来处理该问题。二次元图片描黑边和色块上色的特点，决定其线条的锐利度和色块的纯净度会直接影响观感。为了增加色块区域的平滑度，我们需要进一步对其进行降噪处理。

现有的动漫风格图像超分辨率系统有waifu2x、BigJPG等等。它们的思想都是：先使用超卷积神经网络进行超分辨率，然后使用降噪算法进一步处理图像。

# 实验过程

我们分别尝试了两种超分辨率网络：FSRCNN和VDSR。

## 实验数据

我们没有找到专门为动漫风格图像准备的超分辨率任务数据集，于是我们选择自己收集数据。

### 数据源

由于任务特殊，我们需要确保图片的风格和质量符合要求。我们在插画交流网站pixiv挑选了100张风格相近的动漫风格图片作为训练、测试用到的全部数据。



图 1 收集到的图片

这些图片均满足以下要求：

* 日系动漫风格
* 图片的主体部分是一个动漫人物的面部、半身或者全身特写，并且只出现这一个人物（考虑到这点是因为希望我们的数据集更有针对性。经验表明，绝大部分被作为壁纸的动漫图片，都是针对一个动漫人物脸部或身体的特写）

### 数据预处理

为了限制问题的规模，并在有限的时间内迅速验证模型的有效性，所有图片被维持比例缩小到长边=512像素，另一边长度按比例进行计算。使用了无损压缩（png格式）。

由于小组内两人分别负责两种超分辨率方法，因此数据预处理的其他步骤有所不同。

**FSRCNN**

使用90张数据作为训练集，10张作为测试集，同时用作训练时的验证集。

对于FSRCNN，HR图像和LR图像的边长关系是4:1。训练网络时，网络的输入是10x10的图片块，这些块从LR图像中剪切得到。网络的输出是HR图像中对应位置的40x40的图片块。为了保证网络输入和输出的对应关系正确，具体的实现方法是在本地只保存完整的HR图像。在训练时，从HR图像上剪切下40x40的块，然后降低其分辨率至10x10，获得对应的LR图片块。

测试网络时，直接将整张LR图片放入网络作为输入。该图片可以是任意大小的，因为该网络是全卷积网络，只限制输入图像和输出图像的大小关系，而不限制绝对大小。

训练和测试网络时，数据不进行归一化/标准化。我们也进行过加上标准化的实验，但在将训练后的网络输出的0-1的浮点数恢复到1到255区间时，出现过颜色偏差问题，调试很久未果，估计是运算时精度丢失导致。于是后续实验未进行数据标准化。

**VDSR**

使用HR图像生成对应的LR图像，同样的固定长宽比例，将HR图像缩放到长度为256，作为对应的LR图像。

为了便于处理，将图像转化为对应的灰度图。

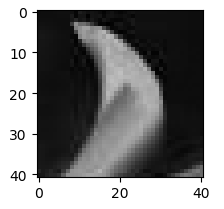
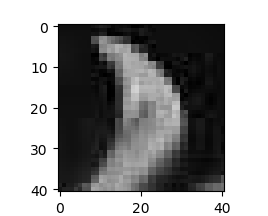
SRCNN论文中将LR图像用双三次插值将LR图像放大到与HR图像相同的大小，裁剪后放进网络中，使用MSELoss计算对应的图像损失。我们同样需要将LR图像放大，实验中我首先使用了双三次插值放大。但是双三次插值放大会在复原的图像中产生网格式的纹路，这一点在实验中影响了网络的效果，因此后来转用了Pillow自带的放大函数，虽然放大的效果不如双三次插值好，但是也不会产生网格纹路。

图 2 双三次插值结果(左) pillow中的resize(右)



将放大后的LR图像作为训练用的输入图像，HR图像作为对应的标签，就得到了VDSR使用的训练数据集。VDSR使用41\*41的图像作为输入，输出同等大小的HR图像。因此将上文的LR放大图像与对应的HR图像裁剪为41\*41的小图像，仍然保持对应的关系，并对数据作图像增强，对图像与标签作水平对称，垂直对称与旋转变换，得到增强后的数据集。VDSR可以使用比SRCNN更高的batch\_size，因此使用batch\_size=128，构造dataloader，传递给网络进行训练。

图 3 输入LR图像(左) 对应的HR图像(右)



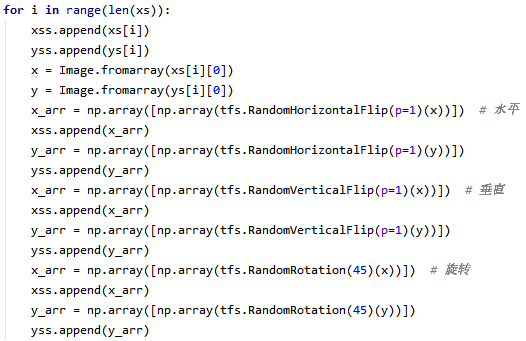


图 4 数据增强部分

## FSRCNN

### 网络结构

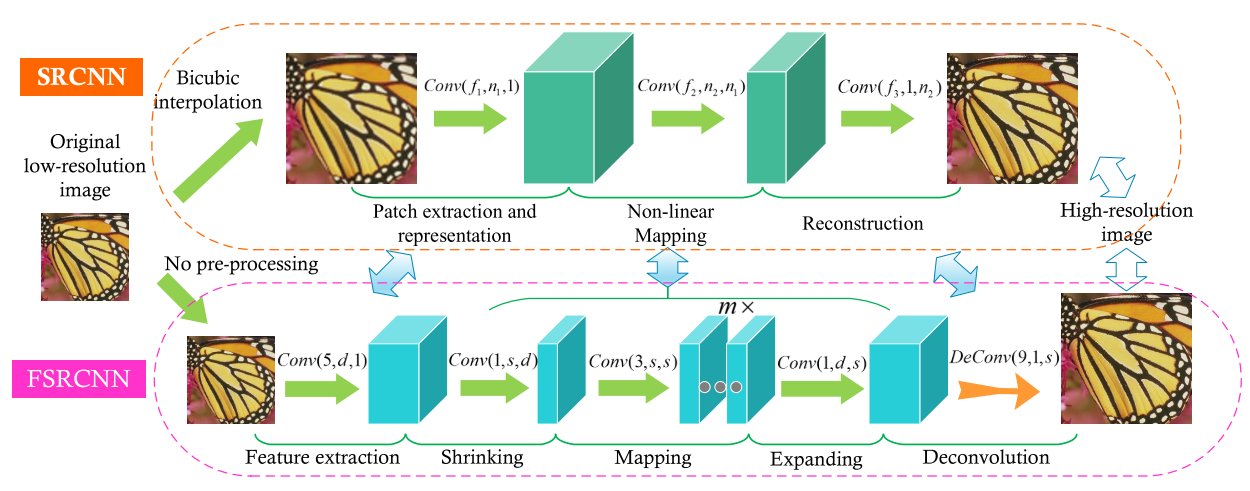


图 5 此图上下两部分为SRCNN和FSRCNN的结构

FSRCNN是SRCNN文章作者对SRCNN的改进，提高了训练和处理图片的速度，做到了实时处理，即每秒处理多于24张图片。

FSRCNN处理图像时，不需要像SRCNN一样进行预处理，直接输入LR图片即可。如图，FSRCNN对图片的处理分为五个阶段。首先是特征提取，使用5x5的filter进行卷积操作，并让feature map具有较高的通道数。Shrinking借鉴了Bottleneck思想，使用1x1的低通道数卷积层来减少网络参数，提高效率。中间的mapping步骤使用了一系列3x3卷积层来寻找一个非线性映射。在前四个阶段，作者让feature map的大小和LR图片保持一致。在最后一步Deconvolution阶段，作者使用一个反卷积操作将中间结果映射成一个和HR图像大小相同的图片。

文章中FSRCNN用来处理单通道图像。但我的实验证明它在处理RGB三通道图像时同样表现良好。将单通道输入拓展到RGB的思想借鉴自SRCNN论文。各种拓展方法之中，结果最优的，也是调整最方便的，就是将网络第一层输入通道数从1直接改为3。

### 训练策略

Loss函数：MSELoss  
优化器：Adam  
Early Stopping：设置EPOCH为2000，最多忍耐300EPOCH验证集表现不下降  
Batch Size：10

学习率：先使用1e-3学习率训练整个网络，验证集表现再无提升时，使用1e-4为学习率对反卷积层进行微调

### 训练过程

1. 使用1e-3学习率训练整个网络的相关指标曲线（实际训练了811个Epoch）

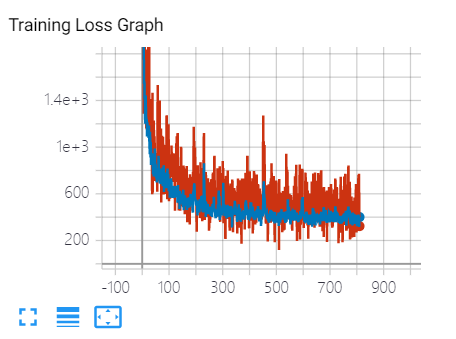


图 7 蓝色是训练集Loss曲线，红色是验证集Loss曲线

图 6 验证集PSNR曲线

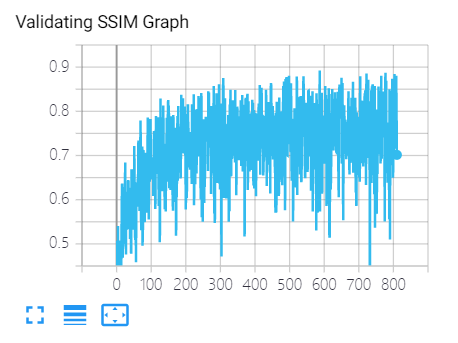
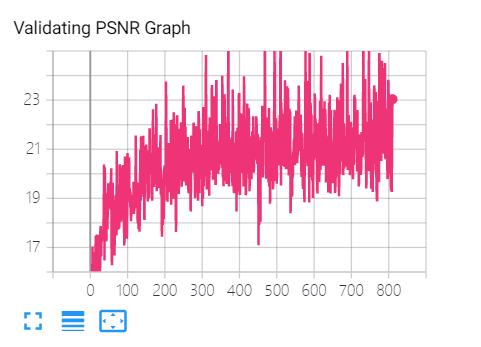


图 7 验证集SSIM曲线

可以看到，训练集合验证集Loss均随训练轮数增多逐渐下降。评估指标PSNR和SSIM得分也逐渐随训练进行而上升。

1. 训练过程中，LR图像、网络的输出和HR图像的对比图（每行顺序为：LR图像、网络输出、HR图像）  
     
     
     
     
     
   可以看到，随着训练逐渐进行，超分辨率网络维持图片中原有结构、颜色的能力也逐渐上升。最终，网络的输出能够较好地在维持几何形状和颜色的情况下，减小锯齿粒度、增大线条平滑度和边缘的锐度。

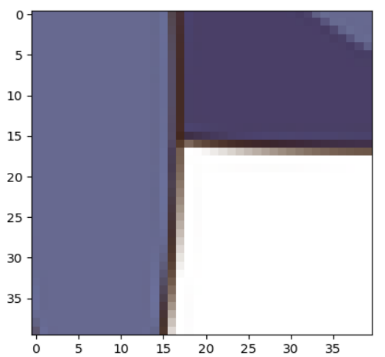
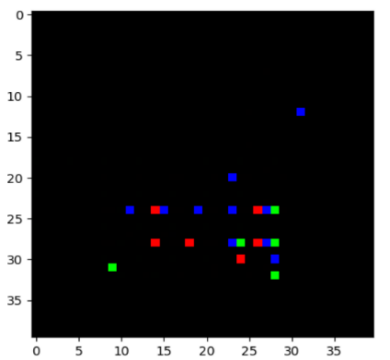
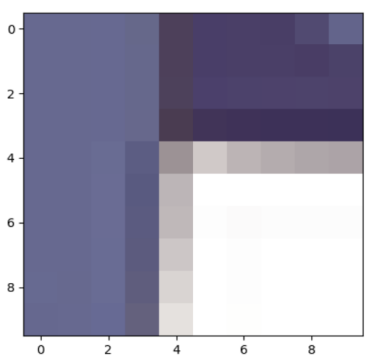


图 8 Epoch 0

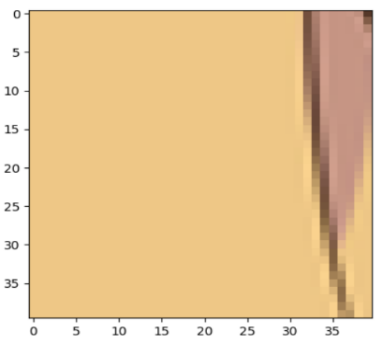
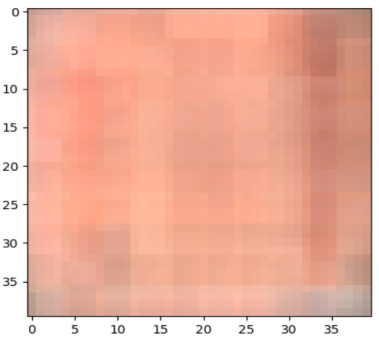
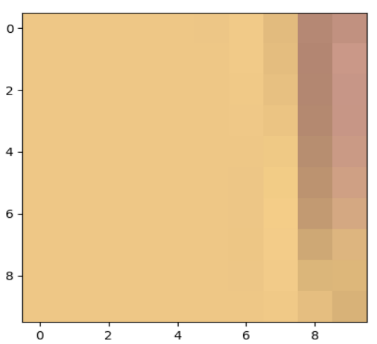


图 9 Epoch 50

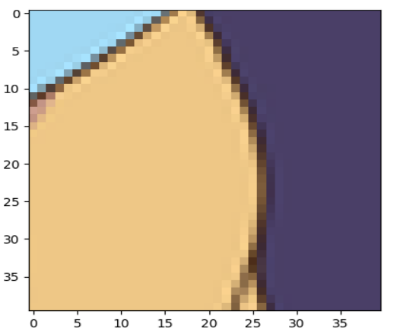
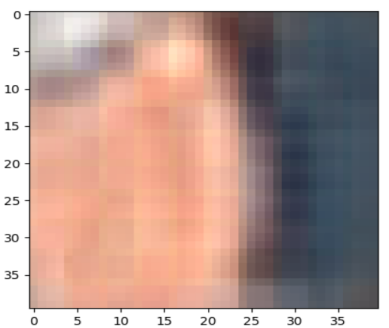
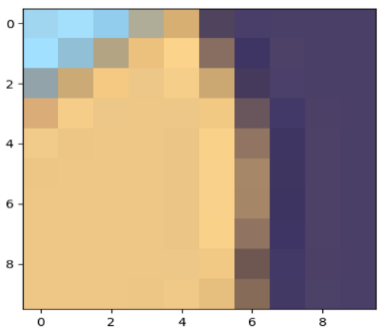


图 10 Epoch 100

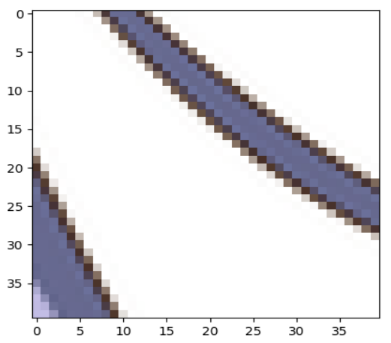
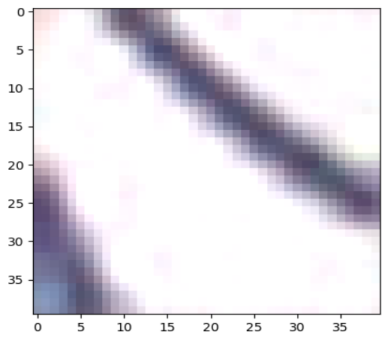
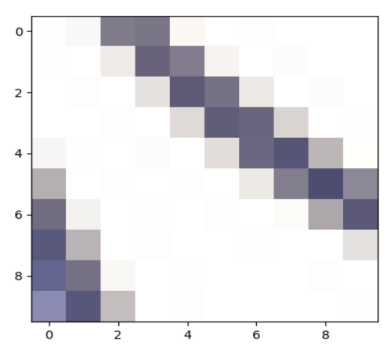


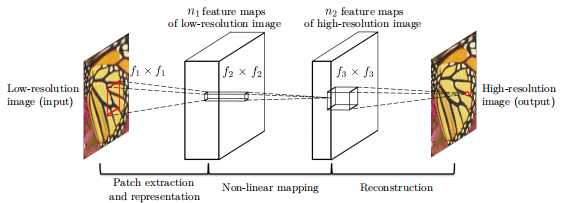
图 11 Epoch 800

## VDSR

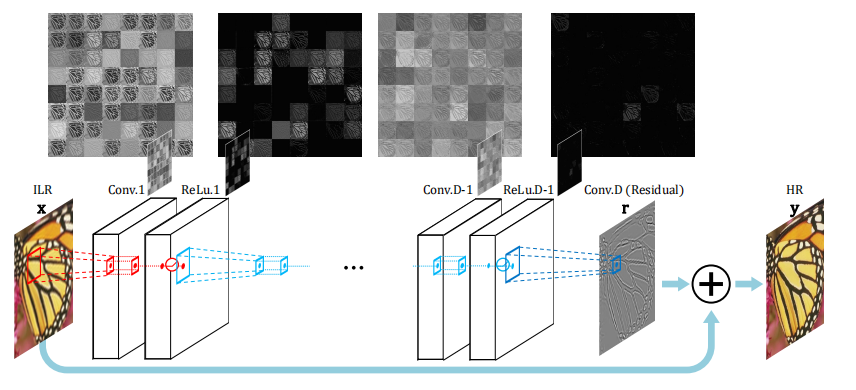
### 网络结构

VDSR使用的网络结构与VGG类似，前18层使用3\*3的卷积层与Relu，输出的特征数为64，添加padding保持输出的特征图与原图大小相同，后面是两层3\*3卷积。一共20层的3\*3卷积层是VDSR结构的主要特征。

VDSR是针对SRCNN而言的，SRCNN的结构比较简单，仅使用了三个卷积层。为了解决SRCNN感受野小，收敛速度慢等问题，VDSR在此之上提升了网络的深度，20层的网络增大了网络的感受野，同时为了加快收敛速度使用了更大的学习率。针对大学习率与深层网络所带来的梯度爆炸问题，VDSR还使用了残差结构与梯度裁剪来解决。

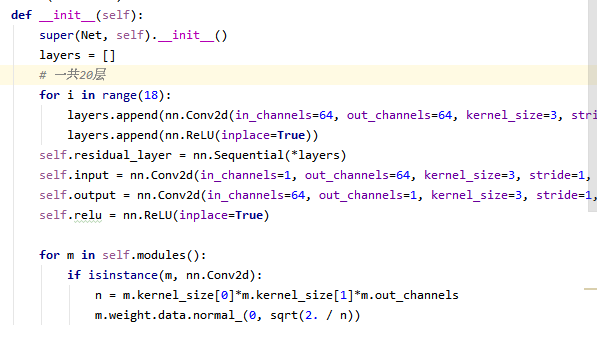


*图12.1 SRCNN的网络结构*

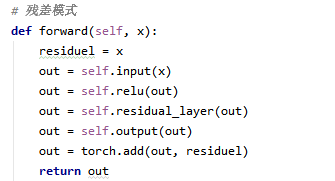


*图12.2 VDSR使用的卷积与残差结构*

对应的网络结构实现如下



对应的残差结构实现如下



在实验过程中，我遇到了网络结构不收敛的问题。具体表现为随着epoch的增加训练集与测试集的Loss一直保持在一定区间内不变。为了解决这个问题，我尝试过简化网络的结构，并使用了14层与8层的VDSR，通过减少前面的层来实现。同样尝试了在网络中添加BN层。

### 训练策略

Loss: MSELoss

优化器: Adam/SGD

学习率调整: lr = lr\*(0.2\*\*(e//10))，每十个epoch变为1/5

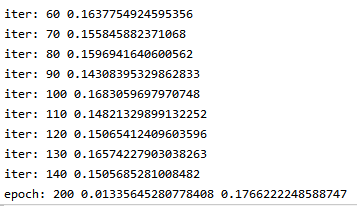
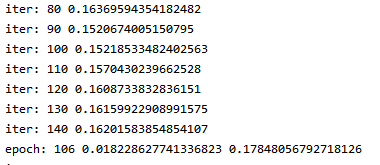
Batchsize=128

梯度裁剪: 0.4

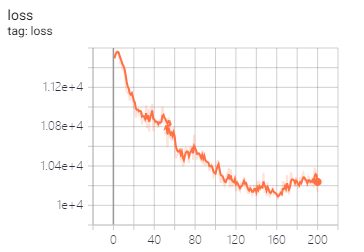
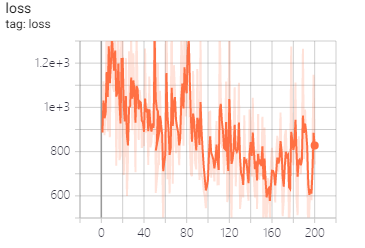
### 实验过程

VDSR的论文中使用lr=0.1的SDG优化器进行训练，并使用参数裁剪以避免梯度爆炸。实验中我为了测试不同的优化器影响，也使用了lr=0.001的Adam优化器与lr=0.05的SDG优化器。以前面的实验的经验来看，Adam一般能取得较好的结果。

但是在实验中Adam并没有能得到更好的结果。训练集的Loss能保持在一定的水平，并没有太大的变化，测试集的Loss更是几乎没有变化。并且Loss只在训练开始时下降了，之后的epoch中几乎没有下降，有时候还出现随着epoch的增加，训练集Loss不变，测试集Loss随着epoch逐渐增加，而后又逐渐下降到最开始的Loss。这种情况下训练结果与最开始的结果几乎是相同的，令人迷惑。



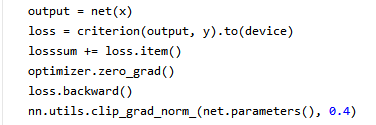
*图13 epoch=106时的Loss(左) epoch=200时的Loss(右)*



*图14 训练集Loss(左) 测试集Loss(右)*

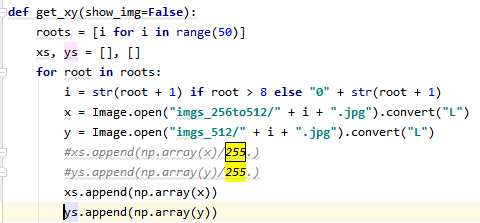
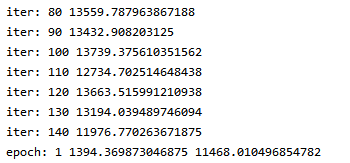
对于训练不动的情况，我也考虑过是否是因为梯度爆炸造成的。因此也设计过lr更小的优化器，以避免梯度爆炸问题，但是结果并没有太多的改善。考虑到最开始训练时Loss下降了一部分，可以判断训练过程中并没有出现梯度爆炸问题。因此需要考虑其他可能的因素。

在实验5中我们使用了WGAN来进行数据拟合，WGAN中同样使用了参数裁剪来避免梯度爆炸问题。同样当网络训练不动时，可以增大clip的值来继续训练。我也考虑过是不是梯度裁剪得太小的问题。把clip的值增大到1.2与4.0后，网络训练并没有出现明显的变化。考虑到网络的参数已经较大了，再增大clip的值并不能明显的改变Loss，因此Loss较小的原因可能并不在参数值的大小上。只能另寻它法。



*图15 参数裁剪部分*

实验中的数据处理部分，我先把图像的灰度范围压缩到(0，1)之内，再送入网络之中进行训练。在考虑到网络的参数可能较小而增大梯度裁剪的clip值时，我同样考虑了用原始灰度值来进行训练，以此得到更大的参数值。出于此考虑我修改了dataloader部分，不进行灰度的01化，而在裁剪成41\*41 的小块后直接送入网络中进行训练。

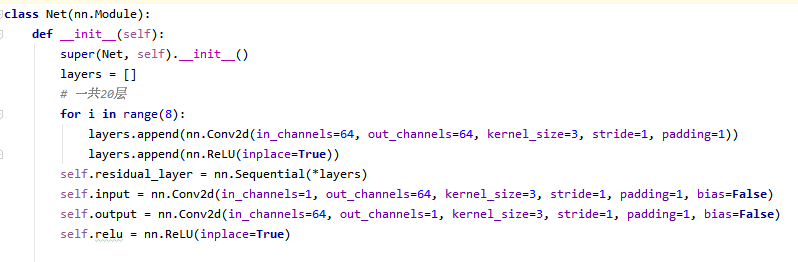


*图16 未01化的Loss(左) 数据处理中的01化部分(右)*

未进行01化的数据Loss很自然的大了几个数量级，但不可避免的是仍然出现了训练不动的情况。虽然如此，但是从实验的结果来看，未进行01化的数据构造的网络处理结果是比进行了01化处理的结果好的。未01化的网络在处理时的数据失真有所减少。

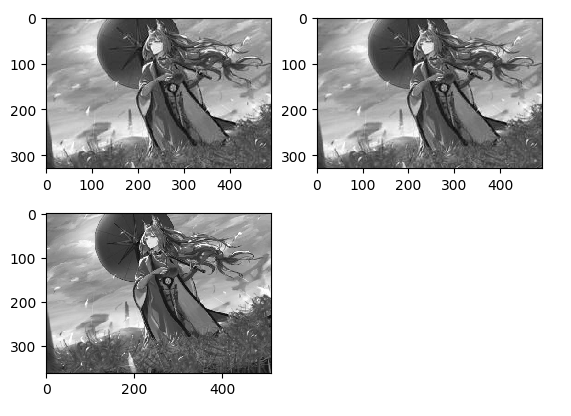
尽管结果仍然不理想，需要进一步的处理。

考虑到网络深度，也有可能是网络训练效果不好造成的问题。即网络层数较深所需要的收敛时间也随之增加。与之对应的解决方法有二。其一是增加训练时间，使用更长的训练epoch数来使模型收敛。这个方向在已经训练了200个epoch仍然没有收敛的情况之下，代价有点大。所以使用另一个方式，尝试简化网络来减小收敛时间。



*图17 一个简化的网络*

实验中尝试了将网络的层数减小到14层与8层，减小了网络之后，训练的Loss也随之减小了不少。但是同样出现了训练不动的问题。增大epoch后也并没有解决。



*图18 LR图像(左上) 10层网络处理结果(右上)HR图像(左下)*

## 去噪

由于时间所限，我们并未使用深度学习方法完成图片去噪，而是使用了传统方法。和FSRCNN一起使用的是Non-Local Means去噪算法。其思想大概是，在图片中寻找和要去噪的图片块相似的图片块，使用找到的相似区域对原区域进行去噪。

# 实验结果分析

## 超分辨率、去噪效果

### 系统的输入输出前后效果对比

**FSRCNN + NLM去噪**

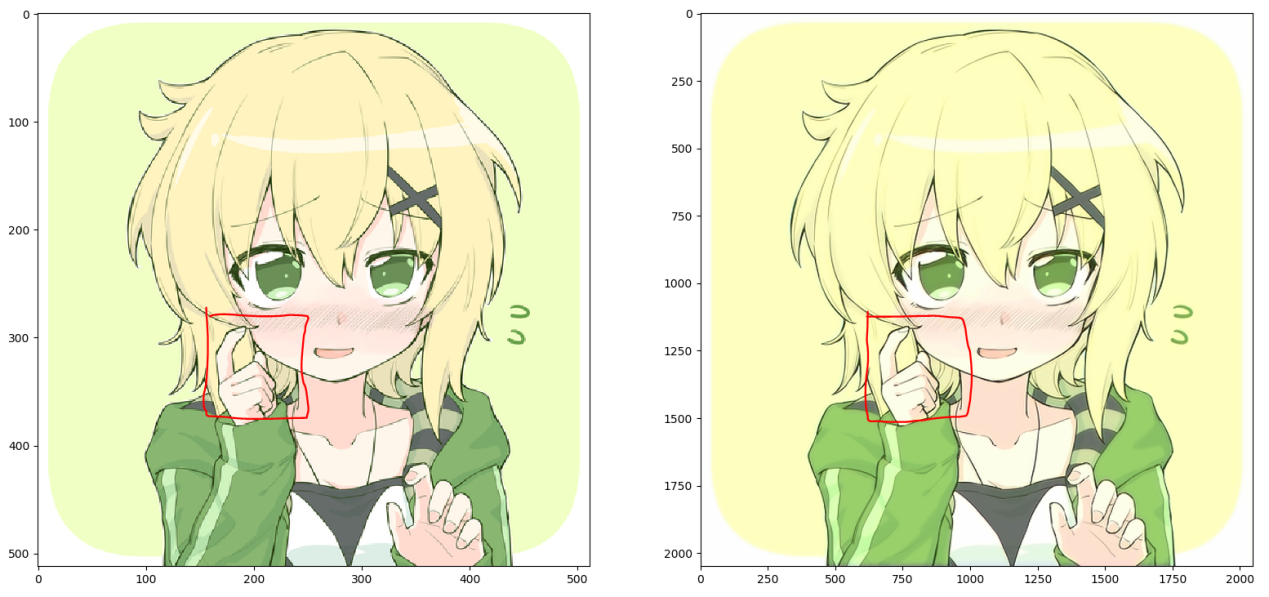


图 19 左图：原图，512x512；右图：使用FSRCNN+NLM去噪，放大四倍。系统输出图片质量明显高于原图，锯齿感大大减少，线条边缘更锐利，线条和色块都更加平滑，图片细节更“精致”。

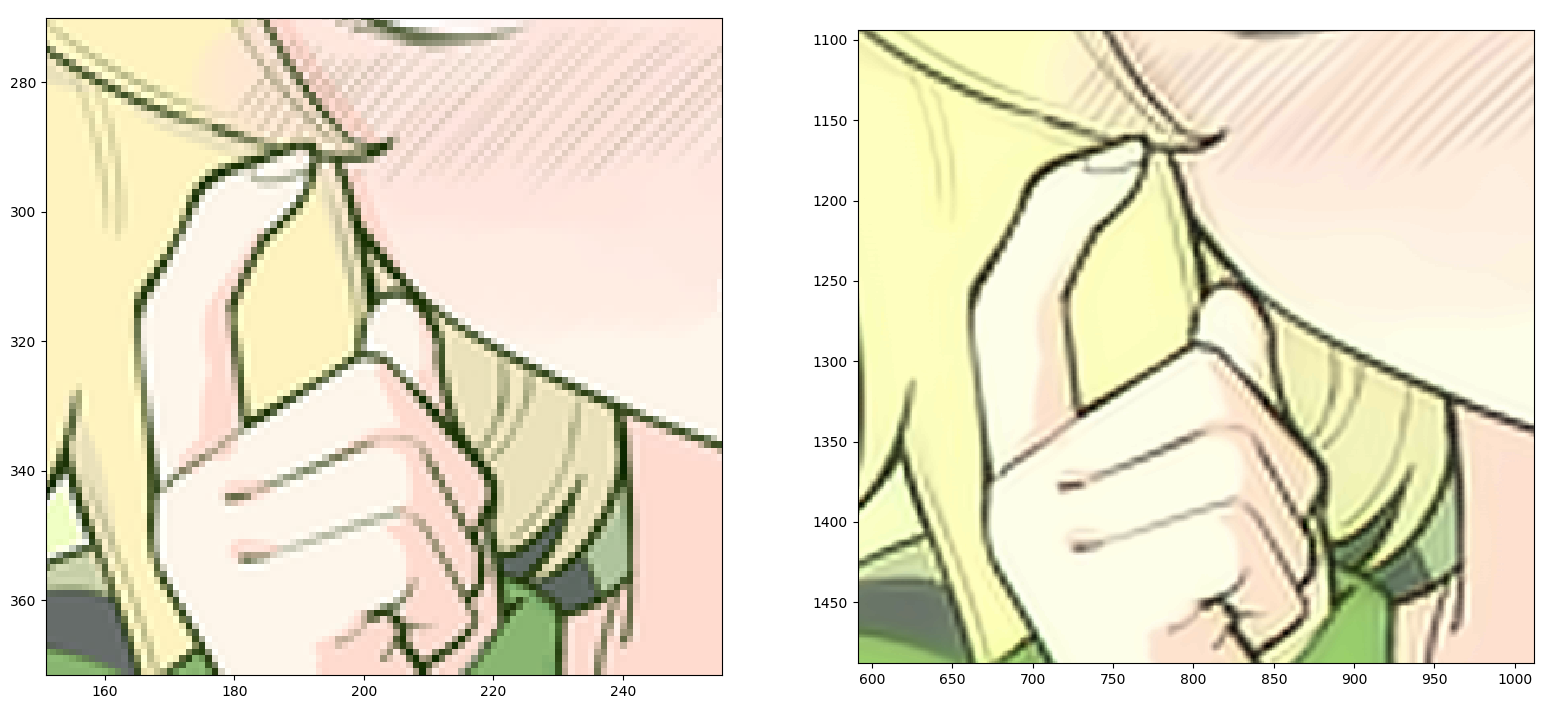
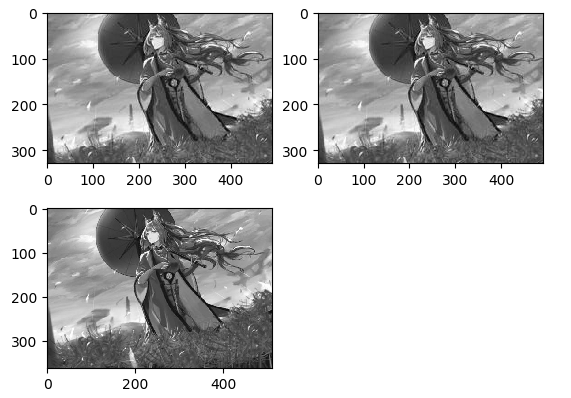


图 20 上图细节对比

**VDSR**

使用训练集中的图像进行图像重建。

由于网络只接收41\*41 的图像输入，所以需要将测试图像裁剪为41\*41的小图像，将多张图像输入网络中进行重建。网络只能对41\*41的图像进行重建。将输出的HR41\*41图像组合起来，就是VDSR的重建结果。



*图21 LR图像(左上) 网络的输出结果(右上) HR原图(左下)*

网络能够做到信噪比的大幅提升，但输出结果并没有能达到肉眼结果的精度提升。



*重建结果(上) 原图(下)*

### FSRCNN和VDSR的重建指标

|  |  |  |  |
| --- | --- | --- | --- |
| 评估指标 | FSRCNN | VDSR | FSRCNN+NLM |
| 测试集平均峰值信噪比（PSNR） | 21.6451 | 26.06 | 21.7169 |
| 测试集平均结构相似度（SSIM） | 0.9175 | - | 0.9198 |

## 有待改善之处

* FSRCNN的输出图片颜色偏亮，怀疑是浮点数计算时精度丢失导致
* 超分辨率网络输出的PSNR指标不及差值法  
  对于FSRCNN网络，训练得到的结果并没有达到论文中30+的水平，原因可能是我们的数据集图片尺寸较小且较为单一，所以超分辨率效果不如论文中的实验。
* 网络的输出比差值法似乎“引入”了更多噪声  
  虽然经过去噪步骤后，大部分引入的噪声都被消除，但我们的实验表明，超分辨率步骤中，插值法会比卷积网络方法得到更“干净”的图像。
* 没有使用深度学习方法进行图像去噪，这一点上仍有提升空间
* 深度网络的训练时间不足，仍存在提升空间。
* VDSR方法虽然有明显的信噪比提升，但超分辨率的实际效果还有待提升。

我们会在接下来的课余时间，从这些方面继续改进本项目。