

目录

1 背景调研.....	2
1.1 超分辨率技术的意义.....	2
1.2 研究内容.....	2
2 理论基础.....	3
2.1 SRGAN 算法介绍与分析.....	3
2.1.1 网络结构.....	3
2.1.2 损失函数.....	4
2.1.3 训练过程.....	4
3 SRGAN 算法改进研究.....	5
3.1 SRGAN 算法分析.....	5
3.1.1 SRGAN 算法存在的问题.....	5
3.1.2 针对措施.....	6
3.2 改进方法.....	6
3.2.1 生成网络的定义.....	6
3.2.2 判别器网络的定义.....	7
3.2.3 损失函数的定义.....	8
4 改进后的 GAN 超分辨率实验.....	9
4.1 操作环境与实验数据.....	9
4.1.1 操作环境.....	9
4.1.2 实验数据.....	9
4.1.3 评价指标.....	10
4.2 对修改后的生成器与判别器的验证实验.....	11
4.2.1 实验步骤.....	11
4.2.2 实验结果.....	11
4.3 对修改后的损失函数的验证实验.....	13
4.3.1 实验步骤.....	13
4.3.2 实验结果.....	13
4.4 将改进后的模型用于手机图片.....	14
5 超分辨率系统设计与实现.....	15
5.1 系统分析.....	15
5.1.1 功能分析.....	15
5.1.2 可行性分析.....	15
5.2 系统设计.....	16
5.3 系统实现.....	16
5.3.1 实现环境.....	16
5.3.2 UI 展示.....	16

1 背景调研

1.1 超分辨率技术的意义

随着计算机与通信技术的高速发展，人们全面进入了信息化时代。人们的知识水平呈现爆发式上涨，数据显示，在我们传递信息的过程中，图像信息占 60%，听觉信息占 20%，因此图像信息的研究在信息化时代具有重大意义。

图像分辨率是体现图像表达能力的重要指标，它所描述的内容是图像含有的像素点个数，也可以说是可度量图像信息量的大小。然而在很多情况下，城乡系统硬件设备受到地域的很大限制，人们无法获得高分辨率的图像。另一方面，网络的受限会导致在传输过程中的图片受损，接收方获取的图片分辨率较低。实际应用中，人们通常想要获取清晰度更高的图像，因此在没有硬件条件的支持下，如何通过更为高效的手段提高图像分辨率是值得深入的领域。

超分辨率算法就是通过对图像的数字信号处理，对一个或多个图像中，重建一组高分辨率的图像，由于此技术不需要硬件支持，在很多领域有着较大应用价值：

（1）医学成像。在医学中获得图像的方式仅仅通过医学仪器，如超声波、CT 成像等。图像的质量也影响医生对病情的把握和判断，因此需要通过超分辨率技术对医学影像的分辨率指标进行提升，更好地帮助医生对患者进行诊断治疗；

（2）公共安全领域。公共安全领域对于视频监控的需求变得极为广泛，例如银行、机场、车站等等。但在一些公共安全安全监控系统中，由于存储容量和布控成本的限制，采集的监控视频信息往往分辨率较低。如果利用图像超分辨率技术能够获得关键帧的高分辨率图像和视频，在出现异常情况时可以提供重要线索来通缉罪犯；

（3）军事侦查。由于硬件技术的限制，使得对复杂的地形、环境等关键信息掌握有限，因此超分辨率技术对侦查领域有着重要应用意义；

（4）图像传输。由于受到网络等因素影响，在网络传输的过程中，只能实现低质量图像或视频的传输，如果超分辨率技术能够得到广泛应用，则用户在接受图像时候，可以通过该技术实现图像分辨率的有效提升，获取更多有价值的图像信息。

图像超分辨率技术在各个领域都有很广泛的应用，本次作业即设计了一种图像超分系统，供以参考。

1.2 研究内容

本课程作业“基于 GAN 的图像超分辨率系统的设计与实现”的研究内容如下：

（1）针对目前的 SRGAN 算法，还存在有着待优化的问题，如细节粗糙、硬件要求高、模型参数数量大等，本次作业针对该系列问题进行一些简单改进，使得超分后的图像能够拥有更多的信息细节；

（2）本次作业设计实现了一个基于 GAN 超分辨率的系统，依托于 PyQt，是一个桌面客户端，可以实现设置输出图片的尺寸，是否需要使用 GPU 等服务功能。

2 理论基础

2.1 SRGAN 算法介绍与分析

使用更快、更深的卷积神经网络来进行单图像的超分辨率的技术已经取得了很大的突破，但是有一个核心的问题目前并没有解决，即我们在对图片进行放大时，纹理细节很难恢复。基于优化的超分辨率方法的行为主要取决于目标函数的选择。目前大部分解决方法集中在最小化均方误差，由此产生的估计值具有较高的峰值信噪比，但是通常缺乏高频细节，而且在某种意义上它们无法与较高分辨率下预期的理想度相匹配，因此在感知上不令人满意。ChristianLedig 等人提出了使用 GAN 进行对单幅图像超分辨率，创立了基于生成对抗网络的超分辨率（SRGAN）。这是人类历史上对图像超分辨率技术上的一个重大突破，能够在 4 倍放大因子下推断图片真实自然图像的框架。主要思想是通过 GAN 来对图像进行重建，修复图像残缺纹理细节，不拘泥于传统的图像质量评价方法，能在肉眼的主观感受下得到提升。生成器通过训练后生成图片，判别器对图片的真伪进行判定，生成器的主要目标就是为了让判别器对生成图像与判别图像难分真伪，这样来实现低分辨率图像转化成高分辨率图像。

2.1.1 网络结构

对于 SRGAN，由三个关键部分组成，分别为生成网络、判别网络和感知损失。生成器 G 是来生成经过训练后的图像的，主体架构是由多个结构相同的残差块来组成，每个残差块有三种操作：卷积、归一化、激活函数，每种操作依次进行，总共进行两次。残差块的主要目的就是图像进行缩放，真正对图像质量进行提升的在最后两个亚像素卷积层。结构图 2.1 所示。判别器的结构图如图 2.2，主要包含的卷积层总数有 8 个，网络逐渐加深的同时特征数由 64 个增长到 512 个。在卷积层之后是全连接层和激活函数层。这里的激活函数使用的是 Sigmoid 函数，这是一种常用的非线性的激活函数，可以将全体实数映射到 $(0, 1)$ 区间上，常用在回归预测上，在鉴别器里面是用来区分获取样本的类别。并且选用 LeakyReLU 激活函数层来代替最大池化层。

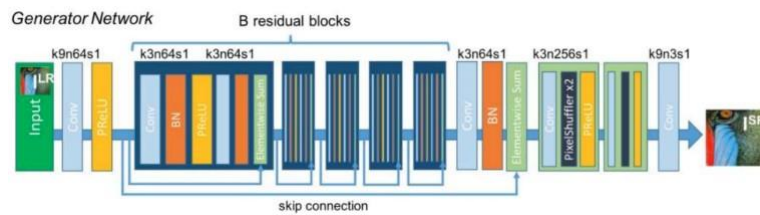


图 2.1 生成器网络结构图

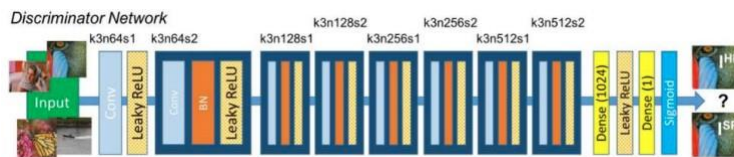


图 2.2 判别器网络示意图

2.1.2 损失函数

损失函数是用来评价模型的预测值与真实值不一样的程度，损失函数越好，则最终训练出来的模型效果越好，当然不同的模型训练效果也不同。损失函数一般分为经验风险损失函数和结构风险损失函数。经验风险损失函数是指预测结果和实际结果的差别，结构风险损失函数是指经验风险损失函数加上正则项。

在 SRGAN 的网络模型中，)损失函数是由内容损失和对抗损失函数进行加权所得，如式 2.1 所示

$$l^{SR} = l_x^{SR} + 10^{-3} l_{Gen}^{SR} \quad (2.1)$$

(1) 内容损失

一般情况下会选择 MSE 作为内容损失，但是经过多次的试验后发现，这会使得最终重建的图像在纹理处理上太过于圆滑。SRGAN 所计算的内容损失函数是通过计算生成器所生成的图像与真正的高分辨率图像二者之间的欧式距离，前提是已知 VGG19 网络中的激活函数层的特征。内容损失函数的表达式如式 2.2 所示。

$$l_{VGG}^{SR} = \frac{1}{w_{i,j} H_{i,j}} \sum_{x=1}^{w_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G(I^{LR}))_{x,y})^2 \quad (2.2)$$

(2) 对抗损失

由于生成器让判别器无法区分数据分布，为了能够让鉴别器分辨能力更强，引入了对抗损失，如式 2.3 所示

$$l_{Gen}^{SR} = \Sigma - \log D(G(I^{LR})) \quad (2.3)$$

其中， $D(G(I^{LR}))$ 表示生成器所生成的图像鉴定为原始图像的概率。使用 $-\log D(G(I^{LR}))$ 是为了对梯度运算来进行改造提速。

2.1.3 训练过程

首先在生成器 G 中输入低分辨率图片，得到高分辨率图片，与此同时将对应的训练集高分辨率图像一同输入到判别器中，输出的鉴定结果生成图像是否为真实图像的概率，如果输出值大于 0.5，则鉴定为训练集中图像，若结果小于 0.5，则为生成器所产生的图像。在模型训练过程中采取生成器与鉴别器交替迭代训练方法。

(1) 训练鉴别器

训练鉴别器的时候，需要暂停生成器模型的参数更新。鉴别器在鉴别生成器生成的图片时，结果接近 0，当鉴别器鉴别高分辨率的结果时，结果接近 1。因此设鉴别器损失值的计算方式如公式 2.4 所示，此后只需调整判别器的网络参数模型让损失值最大化，目的是为了提升判别器的性能。

$$l_D = D(I^{HR}) - 1 - D(G(I^{LR})) \quad (2.4)$$

(2) 训练生成器

训练生成器的时候，需要暂停鉴别器模型的参数更新。计算对抗损失和内容损失，二者通过式 2.1 作为生成器模型损失函数 l^{SR} ，通过调整参数来将 l^{SR} 最小化，达到生成器生成的图像尽可能接近高清晰度图像的目的。

训练示意图 2.3 所示。

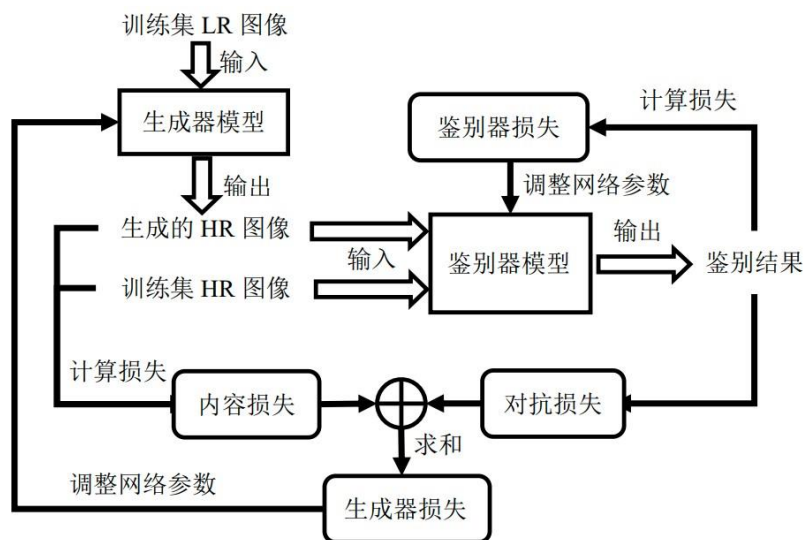


图 2.3 模型训练过程

3 SRGAN 算法改进研究

3.1 SRGAN 算法分析

3.1.1 SRGAN 算法存在的问题

在基于 GAN 的超分辨率图像算法中，其核心思想就是生成器所生成的高分辨率图像尽可能地接近于实际图像，能够躲过判别器的鉴别。对于判别器而言，每迭代一轮，判别器的判别能力会有一定的提升，能够更加准确地区分是真实图片还是生成器所产生高分辨率的图片。由于对于 GAN 模型本身而言，能够生成与实际图像不相上下的样本，而 SRGAN 就是利用了 GAN 所具有的优点，能够生成真实图片经过放大 4 倍后所产生的高清晰度图像，而且更能满足人眼的主观感受，效果好质量高，该方法得到了领域里许多人的认可。



图 3.1 黑洞伪影



图 3.2 真实图片



图 3.3 含有噪声



图 3.4 真实图片

如图 3.1、图 3.2、图 3.3 及图 3.4 所示，SRGAN 目前仍存在很多问题：

- (1) SRGAN 所生成的图片会产生伪影，一般会产生在单一色彩的部分区域，现象大致为黑斑模糊；
- (2) 在细节纹理上，质量有时候会复现波纹、杂乱等，与真实图片出入较大；
- (3) SRGAN 所生成的图片仍有大量噪声，这些噪声并非真实特征，因此影响了图片的真实度。

综上所述，SGRAN 仍有许多需要改进的地方，本次课程作业针对以上存在的问题做出简单改进。

3.1.2 针对措施

(1) 针对 SRGAN 生成的图片会有伪影的问题

一般图片会产生伪影的原因大多是数据集选取不够多样或网络模型中大量的归一化。经查阅相关文献，本次课程作业采用两种方式解决该问题：1、为了避免数据形式单一，尽可能地选取不同的数据集，这样不仅防止了模型泛化能力弱的问题，还有效减少了在训练过程中出现的过拟合情况。2、由于模型中堆积过多的 BN 层，虽然引入 BN 层的目的是加快网络收敛速度，解决过拟合的问题，在图像处理方面被广泛地应用，但是对超分来说，过多的 BN 层会在提取图像的时候对色彩分布造成很大影响，同时对比度也在一定程度会被混淆，因此导致了伪影问题出现。经过对比实验，本次作业决定在生成器的网络模型中将 BN 层删掉，这样就可能减少图片出现伪影的可能性。

(2) 针对 SRGAN 生成的图片存在纹理细节模糊的问题

为了解决生成的纹理细节等问题，主要采取了更改损失函数和生成器的结构。改进损失函数一般包括增加了特征频数较高的损失函数，还有增加了具有判别特征的损失函数。上述所说的改进能够有效增加生成图片的纹理细节质量，提升图像感知度。

3.2 改进方法

3.2.1 生成网络的定义

SRGAN 中的生成器是由提取特征模块以及重建图像模块两部分组成。对于特征提取，一般会采用上采样的方式，上采样最为常用的方法是双三次插值。若对速度有要求，可以替换为双线性插值，但是效果没有双三次插值好。之后将提取特征的低分辨率图像经过 6 个残差块来获得更多的图像纹理信息。在这里经过了多次实验可以证明，图像的特征可以充分通过多个残差块学习得到。这里对残差块进行了修改，去除归一化层。当初引入 BN 层的目的就是获取学习数据分布，因为不同的参数会导致数据分布也不同，因此每次训练迭代，要适应好每次分布的变化，这让网络的训练速度大大折扣，BN 层就会让这些分布收敛速度变快，

可以有效减少梯度的弥散。但是，如果在图像超分上，特征提取会因为 BN 层的存在将色彩归一化，破坏图像本身的空间特征，从而影响最终的图像超分效果。当然，若去掉 BN 层，其计算复杂度还有训练过程产生的内存占用会相应的减少，因此，去掉 BN 层，是个有效的解决方案。改进后的残差块与原图对比如图 3.5 所示。

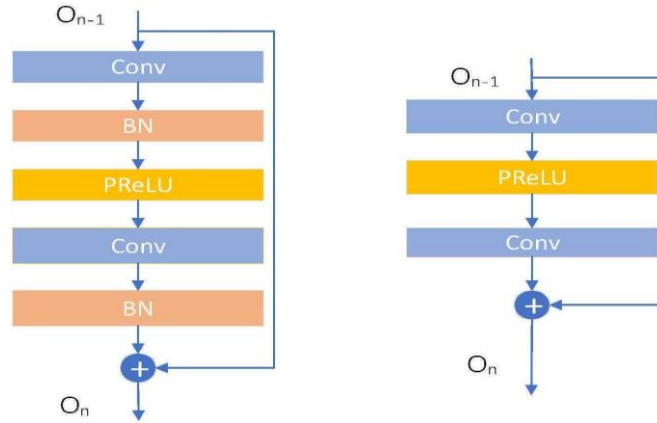


图 3.5 原残差结构与改进后的残差结构

3.2.2 判别器网络的定义

经查阅关于判别器的改进相关资料，Isola 等人所提出的 PathGAN 模型可以很好地解决图像高频细节重建效果处理不够好的问题。

对于原本 SRGAN 中的判别器，输出是经过一系列的卷积后的一个位于 0 到 1 之间的一个标量值，这个标量值是判别器判定生成器所生成图像为真实图像的概率。PathGAN 的思想就是输入的图像分成不同部分 $x_{i,j}$ ，由每个 $x_{i,j}$ 最终映射成一个矩阵 X，这样每一个 $x_{i,j}$ ，可以经过卷积层形成一个 0-1 的数值，所有的数值最终组成一个矩阵。也就是说原本的 SRGAN 所输出的是一个 0-1 的单独的数值，而 PathGAN 输出的是一个包含很多 0-1 数值的矩阵。之所以这样做是因为这样可以对每张图像的具体部分分别进行计算，且都是相互独立的，和整体图像计算相比不仅提高了速度，资源利用率也有了显著提高，可以加深网络的层数，获得更多纹理细节。第二点是可以针对复杂的图片，如图片的各个部分色彩、纹理等细节差异较大，通过对判别器输出的矩阵所有数值最终取平均，这样可以在一定程度上避免纹理细节模糊等问题。

PathGAN 中的判别器又称为马尔可夫判别器，可以用来判别局部块的真假概率值，其网络结构图如图 3.6 所示。

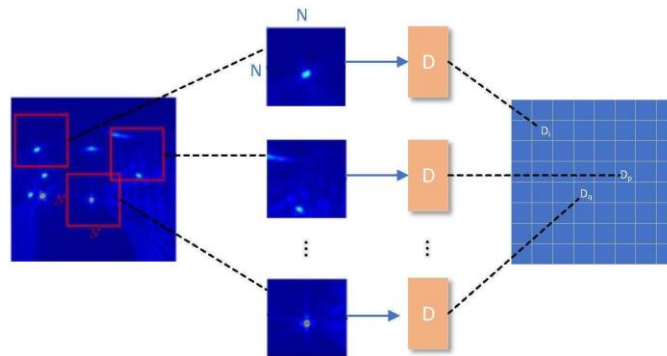


图 3.6 PathGAN 网络结构图

PathGAN 的判别器的主要思想就是可以将图像局部和整体相结合，在恢复图像复杂纹理细节上发挥很大的作用。因此可以将 SRGAN 的判别器改进的思想借鉴于 PathGAN。判别器将输入的图像划分成大小为 $N \times N$ 的不同区域，每个区域经过卷积生成一个概率值，待覆盖完整体的图像区域后，将概率值组成一个概率矩阵，最后将概率矩阵求出最终的均值作为 SRGAN 的输出。在判别器主要经过 6 层的卷积，大小设置为 3×3 ，步长大小设置为 2，卷积层后为池化层、归一层以及激活函数，选取 LeakyReLU 函数作为激活函数，最后链接 sigmoid 函数，此处可以选择每个图像块大小不一样，用到的计算公式如式 (3.1)

$$out_{now} = (out_{pre} - 1) \times s + n \quad (3.1)$$

在式中， out_{now} 表示当前感受野， out_{pre} 表示上一层卷积的感受野， s 和 n 分别表示步长与卷积核大小。判别器的示意图如图 3.7 所示。

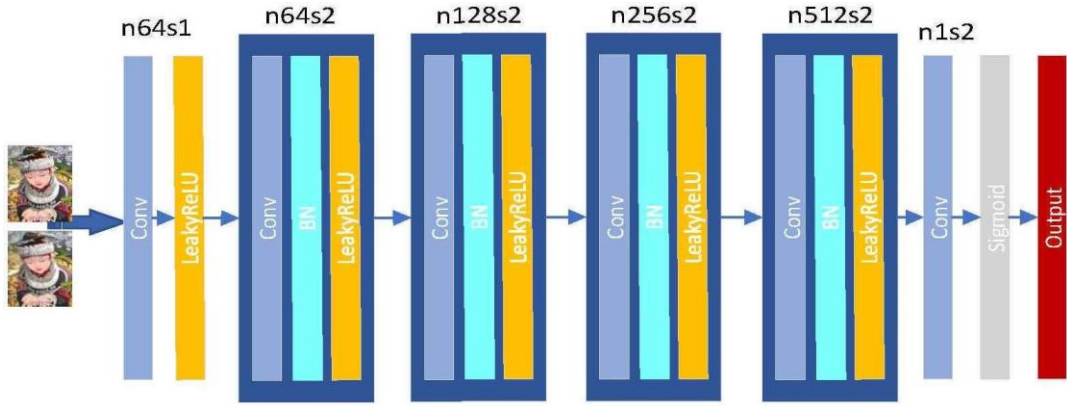


图 3.7 SRGAN 改进后的判别器示意图

3.2.3 损失函数的定义

SRGAN 的最为显著的特点就是提出了损失感知函数，这对图像的纹理细节又有了进一步改善。从 SRGAN 的原文中不难看出，损失函数包括感知损失和正则化损失，而其中的感知损失是由内容损失和对抗损失两部分组成。内容损失主要是指先对生成器所生成的高分辨率图像和原始图像的特征来进行提取，然后再对所提取的特征图进行均方根误差的计算。但是在 SRGAN 中并不是对 VGG19 的每一层网络进行全部提取，而是选择 VGG19 中的一部分层数，比如提取前 5 层特征，或者对 3-7 层特征进行提取。作者针对不同的网络位置进行的实验，分别是 SRGAN-MSE、SRGAN-VGG22、SRGAN-VGG54。

(1) SRGAN-MSE: 直接计算生成器生成的伪图像与真实图像的 MSE 损失。

(2) SRGAN-VGG22: VGG19 网络中第 2 次最大池化前的第 2 个卷积层之前进行特征提取，这定义在内容损失上。

(3) SRGAN-VGG54: VGG19 网络中第 5 次最大池化前第 4 个卷积层之前进行特征提取，定义在对抗损失上，所提取的特征较浅，一般为轮廓等频次较低的特征。

在关于 SRGAN 的论文中，损失函数选取的对比结果如图 3.8 所示。

经过实验发现如果仅以 VGG-54 定义损失函数，会将结果锐化程度过重，这样不仅会导致伪影，产生的噪声也会较为尖锐，若单纯使用 VGG-MSE，会让结果过于顺滑，这样会遮蔽部分具体细节问题。本次作业里结合了每个损失函数的优点进行结合使用，作为联合感知损失，使得生成的图像既满足纹理细节相对平滑，与 VGG-54 产生的噪声相抵消，同时又在人的主观感知上也有较好的视觉感受。阅读相关文献了解到，若图像经过激活函数，其特征会相对稀疏，如果在激活函数后来进行特征提取会导致性能非常弱，因此采用在其前来计算 VGG-22 和 VGG-54 的两种损失函数，并且在其前赋予不同的参数，将内容损失函数改进后，

其公式如式 3.2 所示。

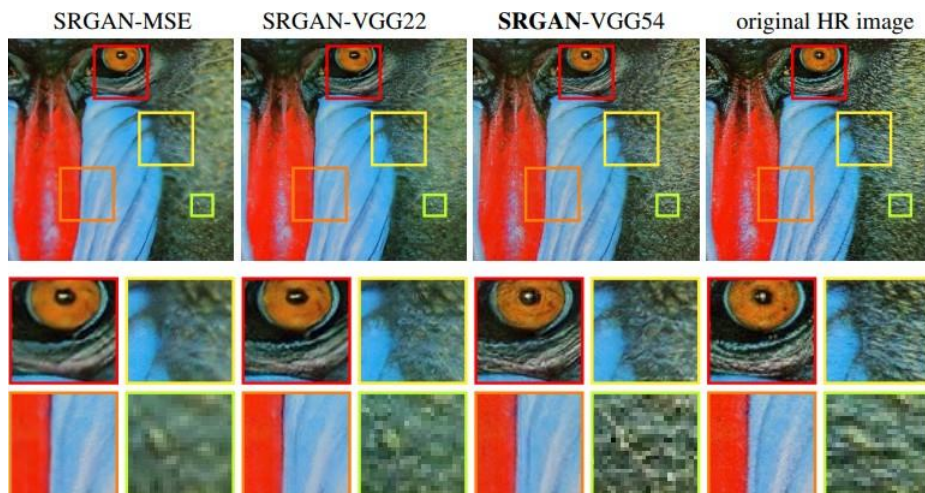


图 3.8 选取不同的网络层的对比图

$$l_{VGG}^{SR} = \alpha \times l_{VGG-22} + \beta \times l_{VGG-54} \quad (3.2)$$

l_{VGG-22} 与 l_{VGG-54} 分别为图像的低频特征和高频特征， α 与 β 为参数，通过实验来确定最终的参数值。最终，本次课程作业选择的损失函数如式 3.3 所示。

$$l^{SR} = l_{VGG}^{SR} + 10^{-2}l_{MSE} + 10^{-3}l_{Gen}^{SR} \quad (3.3)$$

在式 3.3 中， l_{MSE} 为之前所提到的均方差损失， l_{VGG}^{SR} 为对抗损失。为了能够使得生成的图像更加具有在人的主管视觉感知方面的表现力，让生成器生成的图像尽可能地躲过判别网络的鉴别，引入了 l_{VGG} 。

4 改进后的 GAN 超分辨率实验

4.1 操作环境与实验数据

4.1.1 操作环境

实验所选操作系统为 windows11，处理器 AMDR75800H，内存大小 16G。为了让训练时间尽可能缩短，保证实验顺利进行，选取显卡 GTX3050Ti。选用 OpenCV 对数据预处理，选取 PyTorch 为框架作为 GAN 学习模型的构建平台，Python3.6 为训练语言，Pycharm 为编译工具。

4.1.2 实验数据

本次作业在图像的训练集方面选择了 DIV2K 以及 VOC2012。为了对比实验方便，在测试集选择了较为常用的 Set5、Set14 以及自己选择的图像。为了获得图像的低分辨率的样本，需要对数据集先进行预处理操作。预处理的过程如图 4.1 所示。

由于所选的数据集的每张图片的大小都不一样，需要对原始样本进行一系列的预处理过程。首先，先对图像进行分割操作，通过调用 OpenCV 的剪切函数，获取 96*96 大小的图片。然后设定放大因子，根据放大因子确定对图像下采用的参数选择，这里为了保证下采样的质量，选择双三次插值法。以此可以通过原图像获得分辨率较低的图像样本。如图 4.2、图 4.3、图 4.4 及图 4.5 所示。

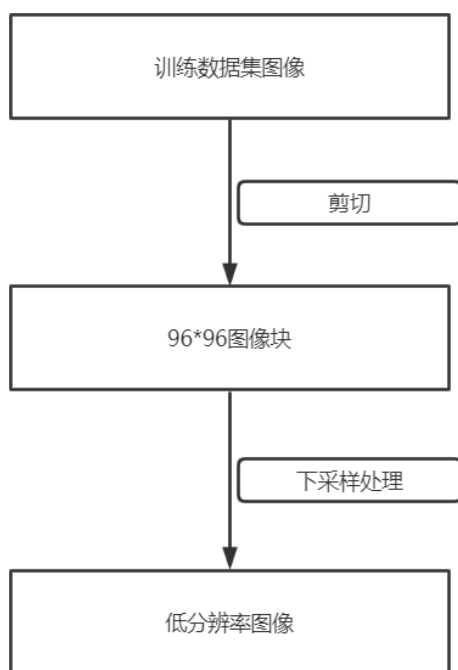


图 4.1 数据的预处理过程



图 4.2 高清原始图像 1



图 4.3 经过预处理之后的图像 1



图 4.4 高清原始图像 2



图 4.5 经过预处理之后图像 2

4.1.3 评价指标

为了能够更加客观地评价改进后算法的性能优势，本次作业选取了 PSNR 和 SSIM 作为评价好坏的关

键参数。其中 PSNR 是基于均方误差来判断原图像与生成图像像素点的差别，MSE 值越低，则 PSNR 值越高，说明生成器生成的图像就更加接近于原图像。而 SSIM 是结合多种元素，如亮度、结构 以及图像对比多方面来对图像质量进行衡量，SSIM 值越高，则说明生成图像的效果越好。

同时用到了 MOS 作为人们主观上的评估指标，来评价不同算法对超分的效果。具体所选的方式是让 7 个人作为该图片的评分员，对图像进行打分，最高分到最低分为 5 至 1，分数越高代表图像越接近于真实图像，即生成的图像质量更高。

4.2 对修改后的生成器与判别器的验证实验

4.2.1 实验步骤

选择 DIV2K 的数据集作为本次实验训练集，但是在实验中可能出现过拟合等问题，将图像统一进行旋转、翻转等操作作为对训练集的拓展。实验的评价标准分为两个部分：主观与客观。主观采用 MOS 评估指标，客观采用 PSNR 和 SSIM 指标进行评价。PSNR 和 SSIM 值越高，代表生成图像与真实图像越接近。本实验将经过预处理后的低分辨率的图像分别输入到原始 SRGAN 以及经过生成器改进后的网络模型中，将两个模型的输出结果的效果图以及评价指标进行对比，具体流程如下：

1、将 DIV2K 的训练集进行剪裁、上采样等预操作，转化成低分辨率图像，再将所转化后的图像输入到模型中；

2、进行特征提取，在第一层进行卷积操作，该卷积大小是 3×3 ，步长是 1，个数为 64，然后经过激活函数的计算；

3、之后输入图像进入残差网络。残差网络有四个组成部分：三个残差块和一个卷积层。由于对残差块经过改进后去掉了 BN 层，因此流程为卷积->批量处理->激活->卷积；

4、得到了残差网络的输出结果后，还需要重复两次操作：卷积和亚像素上采样。这里卷积核的大小和步长与残差网络的卷积核一致，数量上为 256 个；

5、最后进行一次卷积操作，大小为 9×9 ，数量为 3 个，步长为 1。

4.2.2 实验结果

将训练好的模型分别在 Set4 以及 Set14 的数据集上验证，计算改进后的超分辨率算法在预处理后的图像和真实图像客观评价指标与主观评价指标的平均值。

(1) 主观评价结果

选取了测试集中的 5 张图片进行最终的验证。通过对结果的直观对比来看，改进后的模型相较于原本的 SRGAN 模型在细节处理上更为清晰，一定程度上避免了边缘平滑的现象。从图 4.6、4.7、4.8 可以看到，改进后的基于 GAN 超分辨率模型相对于原有基础上，细节处理更加自然、清晰、锐利。在经过了 MOS 评分，几位打分员一致认为改进后的模型所生成的图像质量显著优于原有的模型，说明所提出的改进措施优于 SRGAN 的算法。

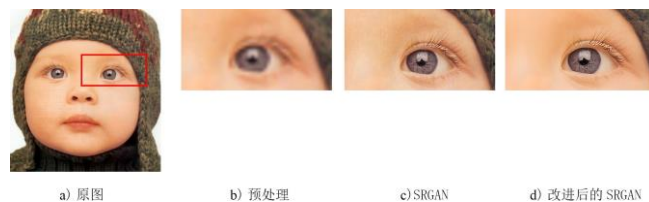


图 4.6 改进后的 SRGAN 在侧视图 baby 局部对比图



图 4.7 改进后的 SRGAN 在侧视图 zebra 局部对比图

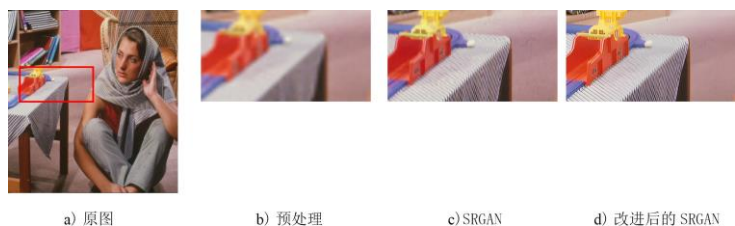


图 4.8 改进后的 SRGAN 在侧视图 block 局部对比图

(2) 客观评价结果

对于客观评价的分析，选取的是平均峰值信噪比（PSNR）和平均结构相似度（SSIM）。在 3 倍的放大因子的情况下，如表 4.1 所示，本次课程作业实验中提出的改进方法在数据集的 PSNR 和 SSIM 指数上均高于原有的模型。为了避免测试结果单一化，将 Set5 和 Set14 所有结果均测试了一遍，最终将结果取平均值，如表 4.2 所示，在数据集 Set5 上，改进后的 PSNR 值比 SRGAN 的高 0.05dB，增长了 2%左右，在数据集 Set14 上，PSNR 值高了 0.02dB，增长了 10%左右。为了更直观地看出改进后每层迭代的数据变化，对每次迭代的模型所生成的图像 PSNR 值进行统计，如图 4.9 所示，改进后的模型在前期与原有模型较为贴合，但在后期逐步超过 SRGAN 模型。因此本实验在 SRGAN 的基础上改进后，无论主观还是客观上，效果在一定程度上都优于 SRGAN。

表 4.1 客观评价指数对比表

Dataset	baby	zebra	block	tiger
PSNR(SRGAN)	30.46	28.32	20.25	26.13
PSNR(改进SRGAN)	31.38	29.54	21.91	27.12
SSIM(SRGAN)	0.81	0.79	0.63	0.77
SSIM(改进SRGAN)	0.85	0.82	0.68	0.85

表 4.2 平均 PSNR 指数对比表

Dataset	原图像	SRGAN	改进SRGAN
Set5	27.43	30.63	30.68
Set14	25.01	27.1	27.33

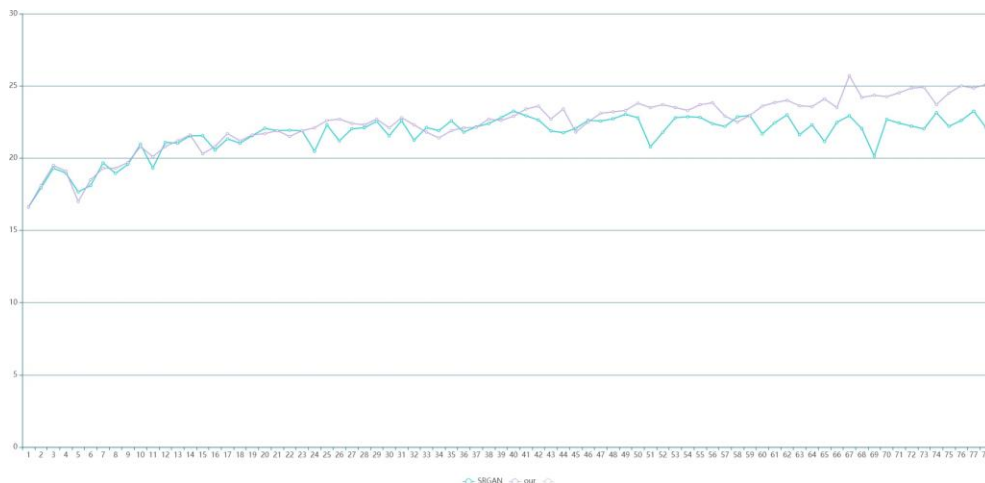


图 4.9 每层迭代后 PSNR 值统计图

4.3 对修改后的损失函数的验证实验

4.3.1 实验步骤

在经过对生成器和判别器结构的改进后，效果相对有一些提升。本实验是为了确定式 3.2 中的特征权重 α 和 β 的最适应的取值，本工作中选取的是 Set5 作为测试集，计算在不同的特征权重下评价指标的变化程度。将 α 选取 0.1、0.01、0.001； β 选取 1、0.1、0.01，共 9 轮实验，评价指标选取 PSNR 和 SSIM。

4.3.2 实验结果

表 4.3 特征值选取 PSNR 值对比

交叉验证	$\alpha = 0.1$	$\alpha = 0.01$	$\alpha = 0.001$
$\beta = 1$	30.43	30.03	29.98
$\beta = 0.1$	30.51	29.84	29.56
$\beta = 0.01$	30.56	29.76	29.29

表 4.4 特征值选取 SSIM 值对比

交叉验证	$\alpha = 0.1$	$\alpha = 0.01$	$\alpha = 0.001$
$\beta = 1$	0.8601	0.8538	0.8509
$\beta = 0.1$	0.8602	0.8470	0.8439
$\beta = 0.01$	0.8607	0.8453	0.8388

表 4.3 展示了不同权重下 PSNR 值的变化，表 4.4 展示了 SSIM 值的变化，可以看到当 α 与 β 值都相对很

小时，内容损失权重与均方误差损失权重占比都很小，在计算过程中，二者的偏重没有被体现出来，这不仅导致两个指标值过小，还会让生成的图像的质量效果也相对较差。当 α 值较大、 β 值较小时，虽然 PSNR 与 SSIM 的值在大小上体现出的优势很明显，但是经过 MOS 评分后发现，在生成图像的具体细节上表现地有所欠缺，人类主观感受较差。当 α 与 β 的值相对较大时，PSNR 和 SSIM 都会取得很好表现，而且高频特征占比更高，在细节处理上会发挥更好的效果。综上考虑各方面数据，最终选择确定 α 值为 0.01， β 值为 1。

4.4 将改进后的模型用于手机图片

为了验证实验不具有偶然性，保证实验结果的真实可靠，在本节将选取手机中一张图片进行验证。将其输入到 SRGAN 模型以及改进后的 SRGAN 模型，将输出结果进行细节对比。



图 4.10 实验图片细节对比

实验效果如图 4.10 所示。对比可以看出，两个模型重建出的图像质量都相对于原图有着很大的提升，花瓣以及绿叶更为清晰，墙上的纹路更为明显。从细节和清晰度来看，改进后的 SRGAN 模型在花瓣处理上更加锐利，清晰度更高。对图像的客观数据对比也发现，改进后的 SRGAN 模型输出图片的 PSNR 值和 SSIM 值均高于原有模型。由此可以说明，SRGAN 对于图像有着提升分辨率的作用，经过部分改进后，图像质量再次得到提升，无论人物、文字线条都具有泛化性，该模型具有可行性。

5 超分辨率系统设计与实现

5.1 系统分析

5.1.1 功能分析

用户在日常使用过程中，有可能会有多张图片同时处理的情况，如果每次只能处理一张图片，会消耗很多不必要时间，因此，系统能够批量处理图像，可以在用户使用效率上得到很大提升。另外，因为独立显卡会让处理过程的速度得到提升，但由于用户的电脑配置不用，有的用户没有可用的 GPU，因此需要用户根据自己的设备进行选择，能够达到个性化的目的。随着网络的发展，单个图像所传递的信息越来越有限，视频的重要性越来越高，因此用户对于视频的超分辨率重建的需求也逐渐迫切，增设视频重建可以满足更多用户需求。此外，因为用户对重建图像的尺寸不相同，系统需要可以对放大因子进行设置。

综上所述，设计出本课程作业系统需要具备以下方面：

- (1) 用户可以上传单张或多张图片，通过该模型获取输出的高分辨率图像；
- (2) 用户可以根据自己电脑配置，选择 CPU 或 GPU 进行训练；
- (3) 用户可以选择所需处理的信息为图像或视频；
- (4) 用户能够自行选择放大因子：2 倍、3 倍或 4 倍，以此设置重建图像尺寸。

本系统的流程图如图 5.1 所示。

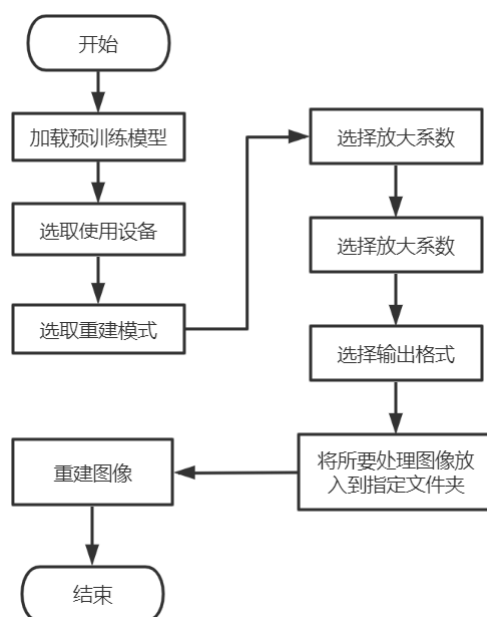


图 5.1 超分辨率系统流程图

5.1.2 可行性分析

(1) 技术可行性分析

本系统主要使用 PyQt 进行系统开发，是可以把 Python 和 Qt 框架二者进行结合的工具集，在 Mac、

Windows 或 Linux 下均可使用。占用内存小，很大保证了运行速度。

(2) 经济可行性分析

基于 GAN 得到图像超分辨率系统可以更加准确直观地处理图像超分的请求。系统总体重建效果好，周期短，花费少，且对计算机性能要求不高，基本都可以使用。

5.2 系统设计

为了能够更加方便地实现界面以及后台业务逻辑的分离，在界面上花费更少的时间，使用了跨平台的 Qt GUI 工具包：Eric。该工具与 QT Designer 结合的很好，在 GUI 程序开发上起到了很大的辅助作用，可以将.ui 文件转化为.py 文件。在界面设计过程中，为了能够使控件跟随窗口大小动态调整，选用布局管理器进行布局。具体实现过程如下：

- (1) 在 QT Designer 中，将所需要的控件拖至设计位置。本系统所使用控件有：按钮（Button）、标签（Label）、清单视图（List View）、文本浏览器（Text Browser）；
- (2) 选择所要布局的窗口控件，使用布局管理器布局，预览布局结果；
- (3) 对每个标签设置属性关系；
- (4) 建立信号和槽的连接，目的是可以让按钮与后台实现通信，通过 `QObject.signal.connect()` 函数进行连接；
- (5) 编写业务逻辑代码，主要包括存读取文件、更改输出格式、视频处理等。

5.3 系统实现

5.3.1 实现环境

完整系统需要两部分来完成，训练部分以及系统调用模型部分。在训练过程中，对计算机的运算能力以及计算机性能有一定要求，配备了 R7-5800 处理器，显卡 RTX3050Ti，具体配置如表 5.1。

表 5.1 系统配置表

项目	具体配置
系统	Windows11
CPU	AMD R7-5800
内存	三星16G
GPU	NVIDIA RTX3050Ti

5.3.2 UI 展示

用户点击该应用，进入如图 5.2 所示界面。

若用户电脑有 GPU，可以选择使用 GPU 提升处理速度，如图 5.3 所示。用户选择重建图像或者视频的放大因子，如图 5.4 所示。当用户选择重建模式为图片模式时，输出格式的选项列表如图 5.5 所示；若为视频模式，则如图 5.6 所示。

开始之前，需在指定文件夹放入需要重建图像，如图 5.7 所示，选取好放大倍数、使用设备等相关参数后，点击运行，执行效果如图 5.8 所示，模型开始工作。

运行结束后，会将结果放入到输出文件夹中，如图 5.9 所示，所有文件格式均为 bmp，而且从图片处理细节来看，图 5.10、图 5.11 重建后的图像 PSNR 值更高，在主观上也能感觉细节处理更好，用户使用方面达到课程作业设计预期。

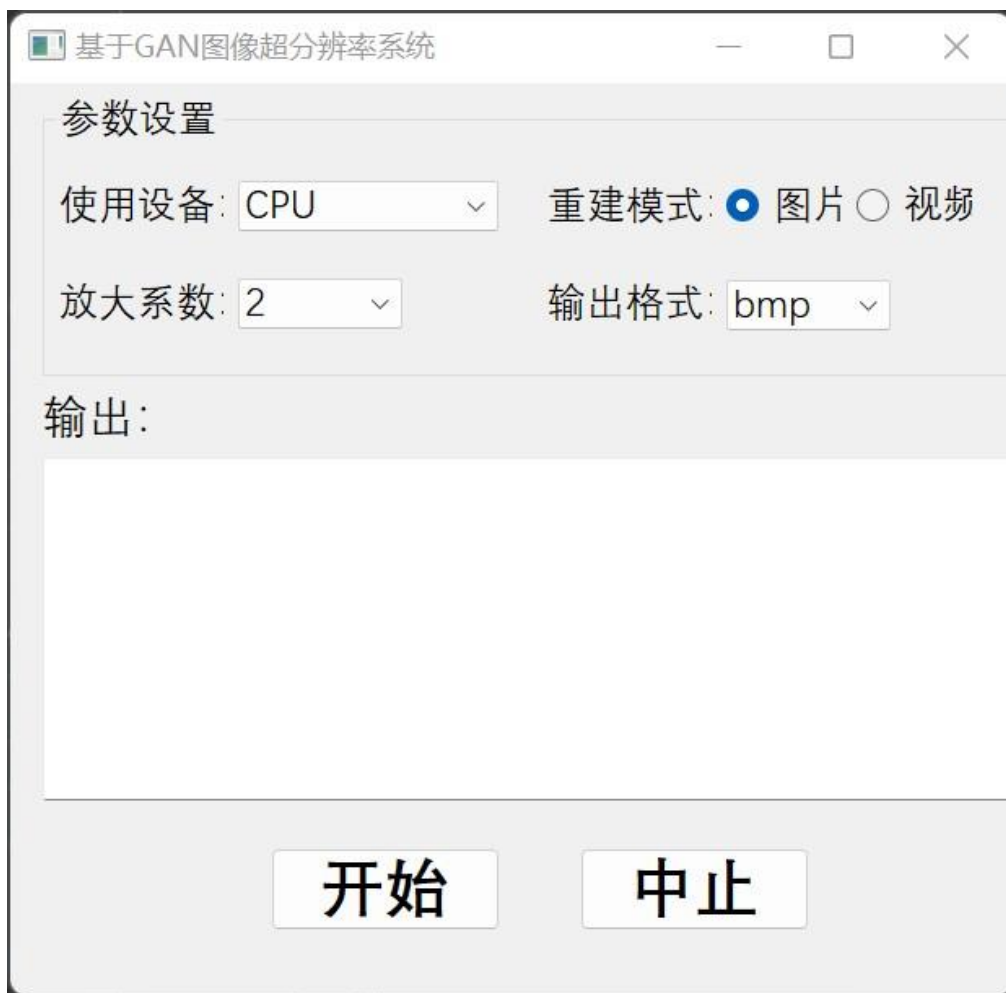


图 5.2 系统主界面



图 5.3 选取运行设备

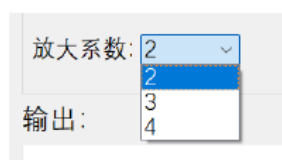


图 5.4 选取重建模式



图 5.5 图片的输出格式

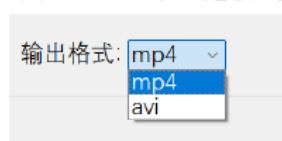


图 5.6 视频的输出格式



图 5.7 输入文件夹图像

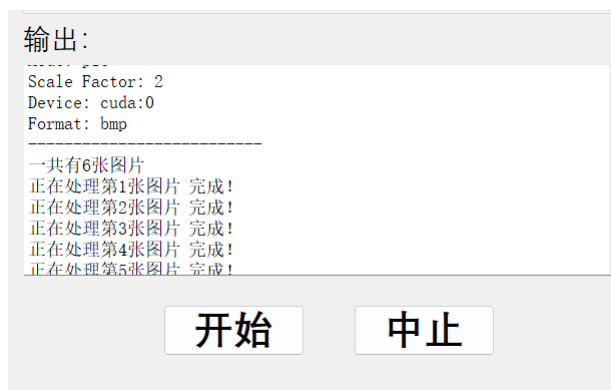


图 5.8 运行时界面



图 5.9 输出文件夹的图像（输出模式选定 bmp）



图 5.10 重建前 (PSNR=27.46)



图 5.11 重建后 (PSNR=29.19)