

计算机视觉与应用

实践

练习3——超分辨率方法

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名:** | 林雅惠 | **学 号:** | 123106222874 |
| **学 院:** | 计算机科学与工程学院 | | |

**2024年5月8日**

目录

[一、 实验原理及步骤 3](#_Toc25892)

[二、 实验方法 4](#_Toc6387)

[2.1数据集加载 4](#_Toc1310)

[2.2网络模型 5](#_Toc11208)

[2.2.1 生成器 5](#_Toc12415)

[2.2.2 判别器 5](#_Toc18856)

[2.2.3 残差块 5](#_Toc5103)

[2.2.4 上采样块 6](#_Toc4291)

[2.3损失函数 6](#_Toc28797)

[2.4训练 7](#_Toc21275)

[2.5测试 8](#_Toc22667)

[三、 实验结果及分析 8](#_Toc26705)

[四、 实验总结 10](#_Toc8434)

# 实验原理及步骤

SRGAN是一种用于单幅图像超分辨率重建的深度学习模型，其核心思想是结合了生成对抗网络（GAN）和残差网络（ResNet）的技术。其算法原理如下:

1. 生成对抗网络（GAN)

GAN包含两个主要组件：生成器和判别器。生成器负责将低分辨率的输入图像映射到高分辨率图像，以实现超分辨率重建。判别器尝试区分生成器的高分辨率图像与真实高分辨率图像之间的差异，即它试图学习生成器输出的图像是否真实。

1. SRGAN的生成器

SRGAN的生成器通常基于深度卷积神经网络（CNN)结构，通常采用残差网络（ResNet）或类似的结构。生成器的目标是学习将低分辨率图像映射到高分辨率的映射函数，为了避免学习复杂的映射，SRGAN的生成器采用残差学习，即学习残差图像（高分辨率图像与低分辨率输入之间的差异）而不是直接学习高分辨率图像。

1. SRGAN判别器

判别器旨在识别生成器生成的高分辨率图像是否真实。它被训练为区分生成器输出的图像与真实高分辨率图像之间的差异。判别器的目标是最大化其正确分类真实和生成图像的能力，从而迫使生成器生成更真实的高分辨率图像。

1. 对抗训练

在训练的过程中，生成器和判别器互相竞争，并通过对抗损失进行训练。生成器的目标是生成足以欺骗判别器的高分辨率图像，而判别器的目标是尽可能准确地区分真实和生成的高分辨率图像。

1. 感知损失

在为了提高生成图像的感知质量，SRGAN还引入了感知损失，通常试用预训练的深度卷积神经网络来计算生成图像与真实高分辨率图像之间的特征空间距离。

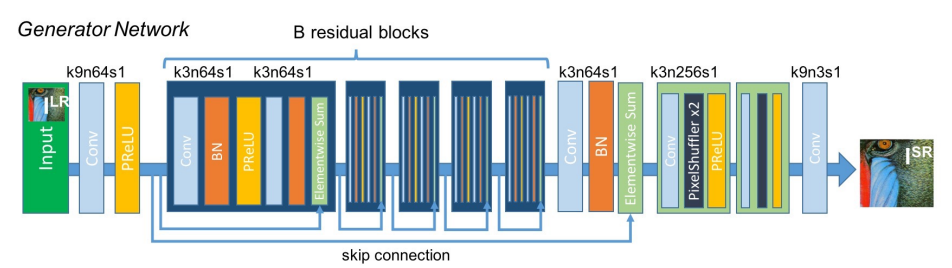


图1

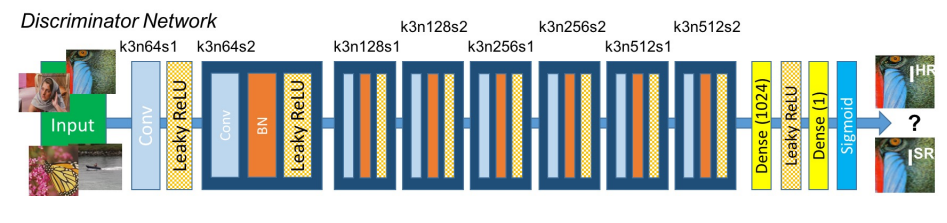


图2

# 实验方法

## 2.1数据集加载

数据集加载中定义了三个类，分别是TrainDatasetFromFolder 类、ValDatasetFromFolder 类和TestDatasetFromFolder类。

TrainDatasetFromFolder 类用于加载训练集中的图像数据，初始化函数中接收数据集目录，裁剪尺寸和放大因子作为参数，在\_init\_函数中，对图像文件进行筛选和计算裁剪尺寸处理，并创建高分辨率和低分辨率图像的转换操作。在\_getitem\_函数中先对高分辨率图像进行转换操作得到低分分辨率图像，然后返回低分辨率图像和高分辨率图像。

ValDatasetFromFolder 类用于加载验证集中的图像数据，与上一个类相似，但在\_getitem\_函数中，返回低分辨率图像、还原后的高分辨率图像和原始高分辨率图像的张量表示。

TestDatasetFromFolder类用于加载测试集中的图像数据，与之前的类相似，但在\_getitem\_函数中对每个图像，通过Resize对象将低分辨率图像进行放大，得到高分辨率恢复图像，返回图像文件名、低分辨率图像、高分辨率恢复的图像和原始高分辨率图像的张量表示。

## 2.2网络模型

网络模型中包括生成器模型、判别器模型、残差块模型、上采样块模型和损失函数模型。

### 2.2.1 生成器

生成器模型中在\_init\_方法中初始化模型的各个组件，计算需要进行上采样的块的数量，定义模型的各个组件，第一个卷积块，包括一个输入通道数为3的二维卷积层，输出通道数为64，卷积核的大小为9，填充为4，以及激活函数。定义五个残差块，每一个块包括一个残差模块。第七个卷积块包括一个输入通道数为64的二维卷积层，输出通道数为64，卷积核大小为3，填充为1以及归一化层。最后一个卷积层，由多个上采样块组成，每个上采样块将特征图的尺寸增加一倍，最后是一个输出通道为3的二维卷积层，卷积核大小为9，填充为4。

在forward方法定义了模型的前向传播的过程，在前向传播的过程中，输入x经过各个卷积层和残差块，最终生成超分辨率图像恶。输出通过Tanh激活函数和缩放操作将像素值限制在0和1之间，以得到最终的生成图像。

### 2.2.2 判别器

判别器模型中在\_init\_方法用于初始化模型的各个组件，定义模型的各个组件，net是一个包含多个二维卷积层、批归一化层和LeakyReLU激活函数的序列模型，每个卷积层后都接了LeakyReLU激活函数，其中LeakyReLU的负值斜率为0.2，输入图像的通道数为3。模型包含多个卷积层和池化层，逐渐减少特征图的尺寸，通过一个自适应平均池化层将特征图转换为1\*1的特征图，最后通过一个卷积层将特征图映射到大小为1的输出，表示输入图像为真实图像的概率。

在forward方法中，定义了模型的前向传播过程，输入x经过各个卷积层和池化层，最终生成一个大小为1的输出，表示输入图像为真实图像的概率，通过Sigmoid函数将其映射到0到1之间。

### 2.2.3 残差块

残差块模型在\_init\_方法初始化残差块中的各个层，self.conv1是一个的二维卷积层，输入和输出通道数都是channels，卷积核大小为3\*3，填充为1，self.bn1是一个二维批归一化层，用于规范化卷积层的输出，self.prelu是Parametric ReLU（PReLU）激活函数，它可以学习激活函数的参数，而不是普通的ReLU那样使用固定的斜率，self.conv2是另一个二维卷积层，与第一个卷积层类似，self.bn2是另一个二维批归一化层。

在forward方法定义了残差块的前向传播过程，首先，将输入x通过第一个卷积层self.conv1进行卷积操作，并通过批归一化层self.bn1和激活函数self.prelu。然后，将上一步的输出通过第二个卷积层self.conv2进行卷积操作，并再次通过批归一化层self.bn2。最后，将输入x与上述输出相加，得到残差块的输出。

### 2.2.4 上采样块

上采样模块在\_init\_方法初始化上采样块中的各个层，self.conv是一个二维卷积层，输入通道数为in\_channels，输出通道数为in\_channels\*up\_scale\*\*2，卷积核的大小为3\*3，填充为1。self.pixel\_shuffle是像素重排操作，它会将输入的特征图重新排列，实现上采样的效果，参数up\_scale指定了上采样的因子，例如，若up\_scale为2，则将每个像素扩展为2\*2的块，若up\_scale为3，则将每个像素扩展为3\*3的块。self.prelu是Parametric ReLU（PReLU）激活函数，用于引入非线性。

在forward方法中定义了上采样的前向传播过程，首先，将输入x通过卷积层self.conv进行卷积操作，得到一个高维特征图。然后，通过像素重排操作，self.pixel\_shuffle将特征图中的像素重新排列，实现上采样的效果。最后，通过激活函数对上采样后的特征图进行非线性处理。

## 2.3损失函数

生成对抗网络中的生成器损失函数中分别定义了总变差损失函数类TVLoss，生成器损失函数GeneratorLoss。

总变差损失函数类TVLoss继承自nn.Module的类，用于计算总变差损失，初始化函数中接收一个输入张量x，计算其在水平和垂直方向上的总变差损失，并返回加权后的总变差损失值。

生成器损失函数GeneratorLoss继承自nn.Module的类，用于计算生成器的总损失，在初始化函数中，首先加载预训练的VGG16模型，并选择其中前31层作为特征提取网络，设置特征提取网络为评估模式，并冻结其参数，定义均方误差损失函数self.mse\_loss用于计算图像之间的均方误差损失，定义总变差损失函数self.tv\_lossde，用于平滑生成的图像。在forward方法中计算四个损失项，包括对抗损失、感知损失、图像损失和总变差损失，其中对抗损失使生成的图像更接近真实图像，目标是最小化生成器对图像的判别结果的平均值与1的差距，如图3所示。感知损失计算生成图像和目标图像在特征提取网络中提取的特征之间的均方误差损失，如图4所示。图像损失计算生成图像与目标图像之间的均方误差损失，如图5所示。总变差损失计算生成图像的总变差损失，用于平滑生成的图像，如图4所示。最后返回四个损失项加权求和的结果作为生成器的总损失。

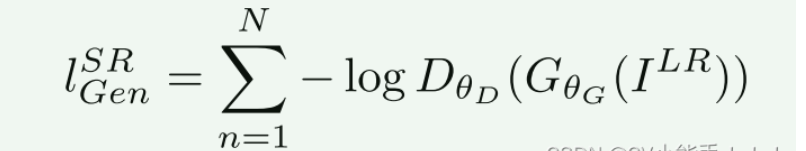


图3

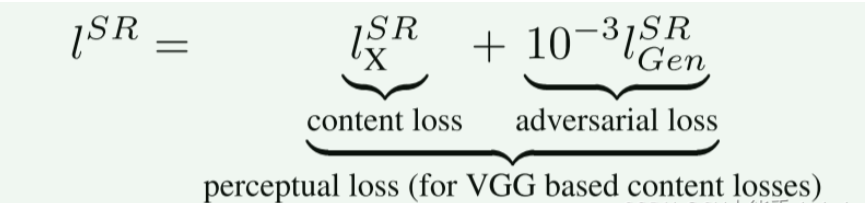


图4

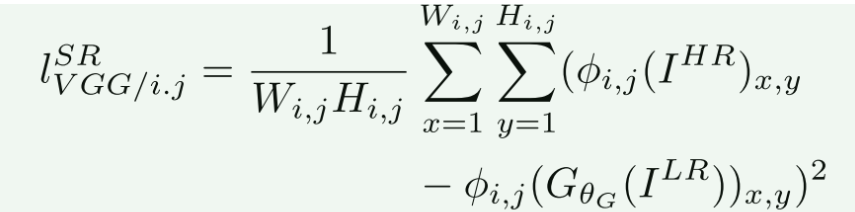


图5

## 2.4训练

首先创建一个argparse.ArgumentParser 的实例 parser，用于解析命令行参数，指定训练图像的裁剪大小、超分辨率放大的因子和训练轮次。接着，加载数据集，根据命令行参数设置裁剪大小，超分辨率因子和训练轮次，创建训练集和验证集的DataLoader对象，用于加载数据。创建生成器模型对象、判别器模型对象和生成器损失函数对象，并定义生成器和判别器的优化器。

在训练循环中，在每个训练的批次中将生成器和判别器设置为训练模式，将输入数据（低分辨率图像）和目标数据（高分辨率图像）加载到变量中，生成器生成高分辨率图像，判别器计算真实图像和生成图像的输出的输出。计算的判别器的损失，更新判别器的参数以最小化判别器的损失。计算生成器的损失并更新生成器的参数以最小化生成器的损失。最后，记录每个批次的损失和分数。在每个训练周期结束后，将生成器设置为评估模式，将验证数据集中的低分辨率图像输入到生成器中，生成对应的高分辨率图像，计算验证数据集上的评估上的评价指标，保存验证结果。最后，保存模型参数和训练的结果。

## 2.5测试

测试过程中，使用ardparse解析命令行参数，包括超分辨率的放大因子和模型的文件名。加载预训练的生成模型，并设置为评估模式。加载测试数据集，包含低分辨率图像和相对的高分辨率图像。遍历测试数据集中的每个图像，使用训练得到的模型生成高分辨率图像，计算生成图像与原始高分辨图像之间的评价指标，包括均方误差（MSE）、峰值信噪比（PNSR）和结构相似性（SSIM），将生成的图像、原始高分辨率图像和恢复后的高分辨率图像保存到指定的路径。

# 实验结果及分析

实验训练得到的模型在Set5数据集上测试时，得到的实验结果如表1所示。测试结果图如图6-10所示。在Set5数据集上测试平均的PNSR为28.48016853379379，平均SSIM为0.8278672218322753，由于SRGAN不是以优化PNSR和SSIM为向导，所以PNSR和SSIM指标值会相对较低。论文中采用训练的数据集是ImageNet，本实验在训练的过程中采用的是VOC2012数据集，论文中提到的数据，损失采用的是VGG22和VGG54，本实验中训练时采用的是VGG16，所以训练得到的模型效果没有论文中提到的模型的效果好。

实验在训练的过程中采用VOC2012数据集进行训练，其中部分图像的纹理和细节相对简单，使得模型重构复杂图像的难度增加，所以在butterfly图像的PNSR指标较低。

表1 测试结果

|  |  |  |
| --- | --- | --- |
|  | PNSR | SSIM |
| Baby | 30.9587 | 0.8474 |
| Bird | 29.9748 | 0.8633 |
| Butterfly | 24.9800 | 0.8526 |
| Head | 29.3184 | 0.7068 |
| Woman | 27.8034 | 0.8735 |



图6



图7



图8



图9



图10

# 实验总结

本实验方法与其它超分辨率方法相比，具有以下优点：

1. 使用了生成对抗网络，通过引入对抗损失，更好地保持图像的真实性和细节，有助于生成更加怕逼真、自然的高分辨率图像。
2. 相比传统的方法，SRGAN生成的超分辨率图像具有更高质量、更细节。 有效恢复图像的细节和结构，使得重构图像更加清晰、自然。
3. SRGAN是一个端到端的模型，可以直接从低分辨率图像生成高分辨率图像，无需额外的后处理步骤，简化了整个超分辨率系统的流程，更好地优化了整体性能。
4. 可以通过调整网络架构和参数来适应不同的应用场景和数据集，具有较强的通用性和可扩展性。

与其它超分辨率方法相比，具有以下缺点：

1. 模型的泛化能力较差：SRGAN对于不同类型的图像可能表现不一致，特别是对于训练数据集中未包含的图像类型，导致模型的泛化能力较差。
2. 训练困难：GAN的训练过程更加复杂和困难，需要花费更多的时间和计算资源来训练稳定的生成器和判别器网络。
3. 对抗训练不稳定：GAN的训练过程可能存在不稳定性。

针对以上提出的确定，以下是可能的改进方向:

1. 针对于模型的泛化能力问题可以通过对任务学习，迁移学习等方法来改进。
2. 针对于模型训练困难的问题，可以通过模型压缩、量化等方法来实现改进模型的计算效率和推断速度问题。
3. 针对于对抗训练不稳定的问题，可以通过改进训练策略、损失函数设计来提高训练的稳定性。