1. 实现说明

代码中网络模型的相关代码均使用了mindspore框架。辅助功能中使用了cv2处理图像和numpy辅助计算。

在我使用的VDSR模型中，它的输出尺寸是等于输入尺寸，而且第一个卷积层的输入通道数为1。这也就说明我们必须单通道的处理图像，因此我在Set5数据集上进行了色彩图和灰度图的双测试。同时在处理图像时使用bicubic插值进行了上下两次采样，使图像回复到原尺寸大小，方便对比超分辨图像和原图像的PSNR和SSIM指标。

首先使用cv2读取图像并进行灰度化，得到原始的色彩图和灰度图。分别使用cv2.resize()方法进行上下采样两次，保持原尺寸不变。然后修改为float类型并将像素值归一化到0-1之间。对于灰度图，需要增加出batch维度和channel维度。而对于色彩图，需要前置通道维度并增加出batch维度，最终图像变为(batch\_size\*C\*H\*W)格式的张量，输入到网络中进行超分辨率处理。然后计算超分辨图像和原图像的PSNR和SSIM指标。

VDSR的模型采用了mindspore框架代码编写。主要涉及mindspore.nn中Conv2d()卷积层和Relu()激活函数的使用。与torch的框架代码中出入最大的是对所有的卷积层进行参数初始化的代码。在torch中，可以使用for m in self.modules()获得每一个模块的参数及类型，而在mindspore框架中，则需要for \_, m in self.cells\_and\_names()和for name, param in self.parameters\_and\_names()两个循环来实现上面的效果。

对于VDSR模型的预训练参数，mindspore与torch也不能达成统一。对于pth文件，mindspore并不支持读取，我们需要重新构建参数字典并保存为ckpt文件，采用让mindspore框架的模型正确读取到预训练参数。

1. 模型细节、优缺点及改进

模型中主要包含多层的卷积层和RELU激活函数的适用以及残差连接的应用。输入图像会经过多次的先卷积再RELU的组合处理。第一次会将图像通道数从1卷积为64，中间的多次使用都会使得图像的通道数保持在64不变。在最后一次中会只使用一个卷积而不是用RELU将图像的通道数重新卷积为1。然后与第一次卷积前的原始图像相加得到最终输出。另外，每一个卷积层的卷积核都是使用的3\*3，并填充1行像素，使得图像在卷积过程中尺寸不发生变化。

优点：相比于之前的方法，卷积层数上的增加，卷积层数直接代表着模型提取特征的能力强弱。小卷积核的进一步引入，利用卷积核3x3堆叠层数，模型一共20层。引入了残差网络。

缺点：原文依然采用的MSE损失，单纯比较像素之间的差异。图像相对比较平滑。训练依然采用是SRCNN的训练方法，先上采样到高分辨率尺寸大小再进行训练。

改进方法：结合多尺度的信息来进行超分辨率重建，以提高模型对不同尺度图像的适应能力和重建质量。

1. 运行结果及说明
2. 运行结果



从左向右依次为原图像，下上采样后图像，超分辨率图像

超分辨图像的PSNR和SSIM值为32.46969509378902、0.9275833348025603

1. 结果说明

代码运行结果正常，可以得到超分辨率后的图像。从上面的示例中，尤其是婴儿的瞳孔光亮中可以明显的看出，超分辨图像相较于原图像和采样图像更为生动。