

论文《Variable-Rate Deep Image Compression through Spatially-Adaptive Feature Transform》复现报告

摘要

这篇论文提出了一种基于深度学习的图像压缩网络，通过引入空间特征变换（Spatial Feature Transform, SFT）模块，实现了基于质量图的可变速率图像压缩方法。该方法能够在单一模型中覆盖多种压缩率，并支持任意像素级别的质量控制，从而实现空间自适应图像压缩。这一框架不仅提高了压缩性能，还允许通过优化目标任务特定的质量图来实现任务感知压缩，如分类任务。特别地，该方法不需要为不同任务单独训练模型，而是利用预训练的网络进行质量图优化，显著提升了分类性能以及文本区域质量的保留。实验结果表明，该方法在比特率和失真之间实现了优异的权衡，与传统的固定比特率模型相比，能够在更广泛的任务中取得更好的表现。此外，该模型在不同任务（如分类任务和文本保留任务）中表现出显著的优势，通过自动生成的质量图在目标任务上实现了显著的性能提升。通过与传统方法的比较，该方法在可视化质量、峰值信噪比（PSNR）和多尺度结构相似性（MS-SSIM）等指标上均取得了领先。

关键词：空间特征变换；可变速率图像压缩；任务感知压缩；质量图。

1 引言

图像压缩是计算机视觉和多媒体技术领域的核心问题之一 [1]，其目的是在保证图像质量的前提下最大限度减少存储空间和传输带宽。随着移动设备的普及和社交媒体的兴起，大量图像数据的存储和传输需求日益增长，对高效的图像压缩技术提出了更高要求。传统的图像压缩方法，如 JPEG、JPEG2000 和 BPG，尽管在一定程度上满足了应用需求，但由于其依赖于手工设计的特征提取与压缩策略，在处理复杂图像时常出现性能瓶颈。

近年来，基于深度学习的图像压缩方法逐渐兴起。这类 [2] 方法通过端到端优化框架自动学习图像表示，能够显著提升压缩效率。然而，大多数深度学习方法仅适用于单一压缩率，且无法灵活适配不同任务场景的需求。尤其在涉及关键区域保留或任务特定要求（如分类或目标检测）的应用中，传统压缩模型难以兼顾全局性能与局部重要性的差异。

论文《Variable-Rate Deep Image Compression through Spatially-Adaptive Feature Transform》[7] 提出了一种基于空间特征变换的变率图像压缩网络，提供了新的解决方案。该方法通过引入质量图，将像素级重要性融入图像压缩过程，从而实现更高效的比特分配。相较于传统方法，该模型能够在单一网络中实现多种压缩率的支持，同时通过质量图的优化在无额

外训练的情况下实现任务感知压缩。这种灵活性和高性能使其在图像存储、传输以及特定任务处理领域具有广阔的应用前景。研究这一方向具有重要意义。首先，基于深度学习的变率图像压缩方法体现了下一代图像压缩技术的发展趋势；其次，探索任务感知压缩的实现路径，不仅能满足多样化的场景需求，还可进一步提升计算机视觉系统在资源受限环境下的性能。

本论文的选题意义体现在三个方面。首先，从理论意义上看，变率图像压缩技术通过深度学习与像素级优化策略的结合，提供了一种新的理论框架和实现途径，推动了图像压缩领域的技术创新。其次，从实际价值来看，该技术在云存储、智能手机和视频监控等应用中，能够高效地压缩图像并满足不同任务需求，为数据传输优化和存储管理提供了全新的解决方案。最后，随着 5G 和物联网的快速发展，图像数据量急剧增加，研究更高效、更灵活的图像压缩技术不仅能显著降低数据传输和存储成本，还对社会的数字化进程具有积极的推动作用。通过对选题背景、依据与意义的深入分析，本文旨在系统性地研究变率图像压缩技术，探讨其在不同场景下的应用潜力，为图像处理领域的技术进步做出贡献。

2 相关工作

2.1 传统图像压缩

传统的图像压缩方法通常基于信号处理和手工设计的特征提取策略，其核心目标是通过数学模型和编码算法有效减少数据信号。这些方法可以大致分为以下几类：

1. 基于变换的压缩方法（如 JPEG、JPEG2000）：通过离散余弦变换（DCT） [3] 或离散小波变换（DWT） [6] 将图像从空间域转换到频域，对频域函数进行量化和编码。DCT 在 JPEG 中被广泛应用，而 JPEG2000 采用 DWT 以支持多这些方法的压缩性能较好，但在高压压缩率时容易出现失真伪影，如块效果。

2. 区域感知压缩（如 BPG）：基于对感兴趣区域（ROI）的指定或手动设计的掩码，提供对不同区域的差异化压缩质量支持。这种方法 [4] 通过牺牲非重要区域的质量来提升重要区域的重建效果，但通常仅支持固定质量等级的区域划分。

3. 熵编码（如 Huffman 编码、算术编码）：通过为不同概率的符号分配变长码字实现信息量的最小化。此类 [5] 编码方法常与变换压缩结合使用，作为最后的编码步骤。

2.2 基于空间特征变换的深度图像压缩

论文 [7] 提出了一种基于空间特征变换（SFT）的深度图像压缩网络模型，该模型能够处理不同的压缩速率。该网络接收源图像和对应的质量图（Quality Map）作为输入，生成具有可变压缩速率的模型，并通过任意像素级质量图调整压缩效果，模型可以适应不同的压缩需求。以下为本文方法的主要分类与描述：

1. 变率压缩：通过像素级别的质量图实现空间自适应压缩，不同像素的比特率依据其重要性动态调整，从而实现高效的比特分配。相比传统的单一压缩率模型，本文方法在一个模型中支持从低到高的多种压缩率。

2. 基于深度学习的端到端训练：采用卷积神经网络作为编码器和解码器，通过优化率失真损失函数实现端到端的压缩训练。本文的网络包括空间特征变换（SFT）模块，该模块结合图像特征和外部质量图进行像素级的特征变换。

3. 任务感知压缩：通过优化目标任务（如分类或检测）的特定质量图，增强相关区域的压缩质量。这种方式无需为不同任务重新训练模型，能够显著提高压缩后图像在特定任务中的性能。

3 本文方法

3.1 本文方法概述

此部分对本文将要复现的工作进行概述，本文 [7] 提出了一种基于深度学习的变率图像压缩方法，利用空间特征变换（Spatial Feature Transform, SFT）模块，通过像素级别的质量图实现空间自适应压缩。该方法的核心创新点在于同时支持多个压缩率，并能根据目标任务生成特定的质量图，在无需重新训练模型的情况下，优化特定任务的压缩效果。

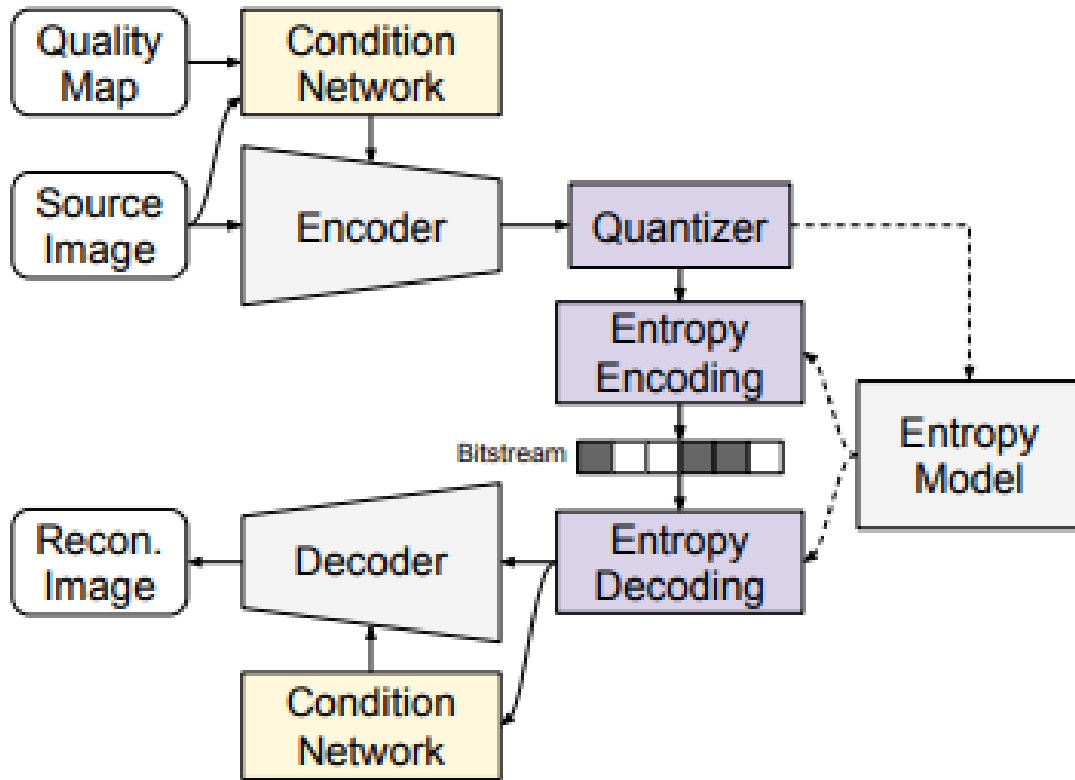


图 1. 方法示意图

具体而言，该方法包括以下三个关键部分：

1. 基于质量图的变率压缩：基于质量图的变率压缩方法通过引入像素级别的质量图，实现对图像不同区域的差异化压缩控制。质量图作为二维矩阵，与图像尺寸一致，其每个像素值表示该位置的重要性，高值区域在压缩中分配更多比特以保证高质量重建，而低值区域则减少比特分配以节省存储空间。在编码阶段，质量图与图像同时输入，通过条件网络生成像素级调节参数，并结合空间特征变换（SFT）模块动态调整特征的重要性，生成潜在表示后进行量化与熵编码。在解码阶段，通过潜在表示的辅助信息恢复质量图替代，配合逆向特征变换实现图像的高质量重建。通过优化率失真损失函数，模型在比特率和重建质量间取得优异

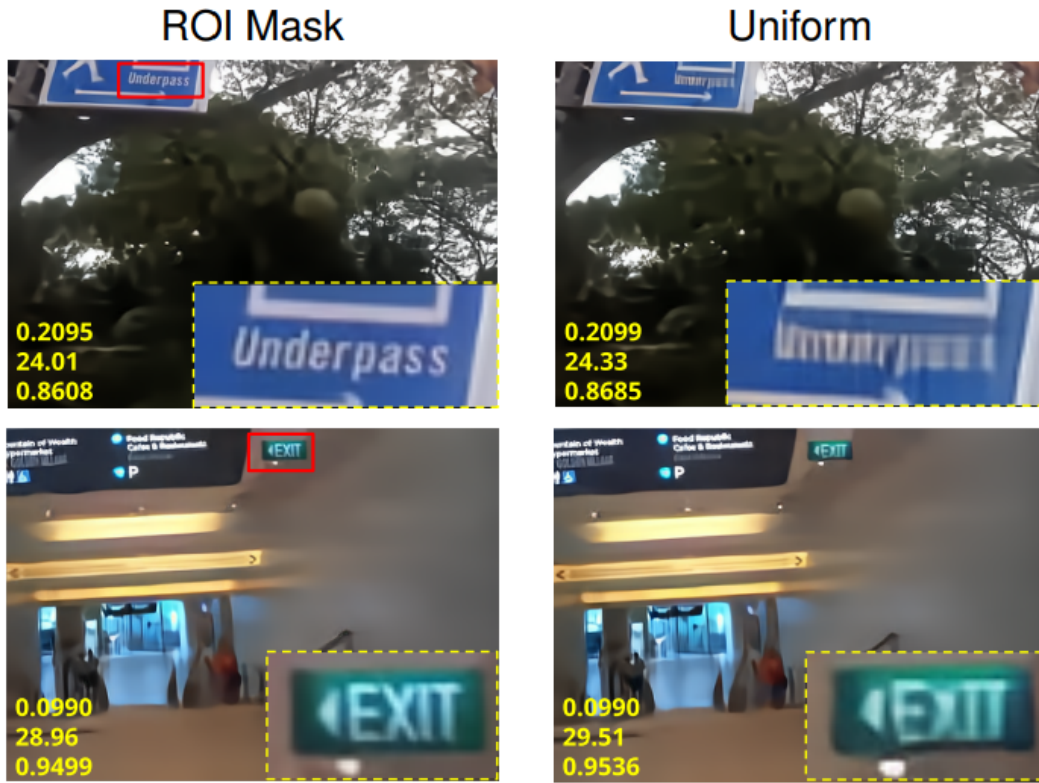


图 3. 基于任务感知的图像压缩与统一质量图的图像压缩对比

3.2 特征提取模块

特征提取模块是基于质量图的变率压缩模型的关键组成部分，负责将输入图像转换为潜在表示，同时结合质量图实现像素级的特征调节和比特分配。模块的主要组成包括编码器、解码器和条件网络。编码器通过一系列卷积层提取图像特征，并结合输入的质量图生成潜在表示。质量图通过条件网络生成像素级的调节信息，作为空间特征变换（SFT）模块的输入，对特征进行仿射变换。这一过程确保了重要区域特征的增强和非重要区域特征的压缩。解码器在重建图像时通过逆向过程恢复特征，同时结合辅助信息（如从潜在表示生成的替代质量图）进行解码，确保空间自适应重建质量。条件网络作为特征提取模块的核心，动态生成输入特征和质量图的联合条件信息，用于指导编码器和解码器中的特征变换。整体而言，特征提取模块通过深度卷积神经网络与 SFT 模块的结合，实现了像素级动态特征调节和灵活的比特分配，为后续的量化与熵编码奠定了基础。

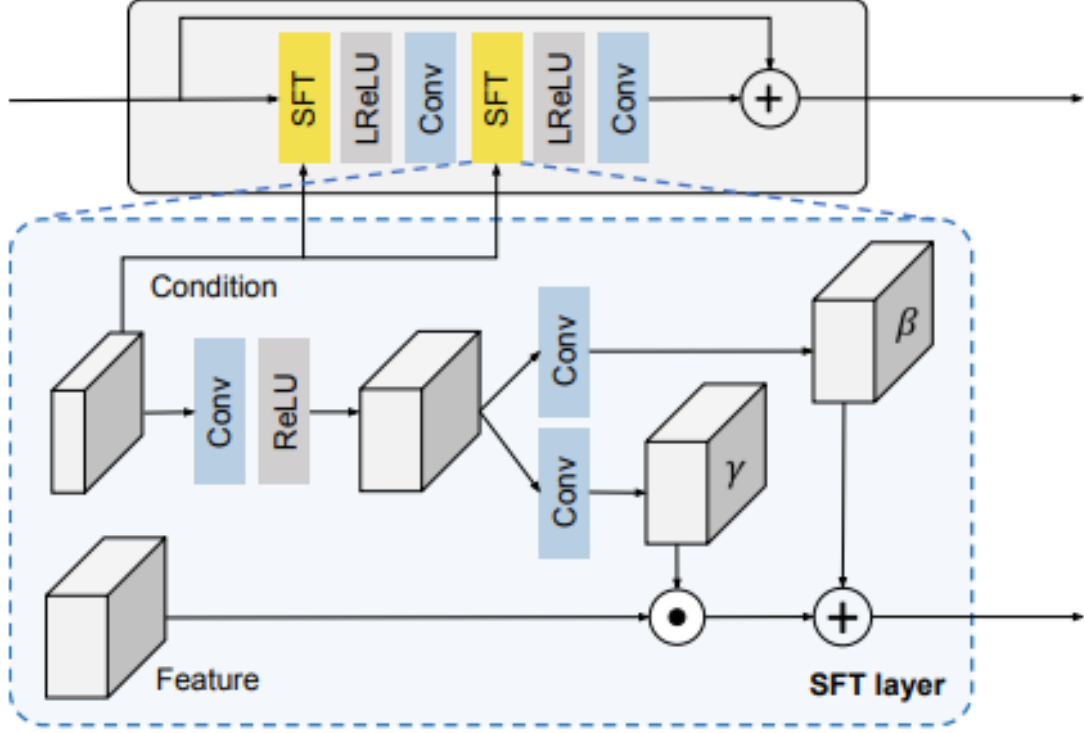


图 4. 特征提取模块示意图

3.3 损失函数定义

本文通过优化率-失真（Rate-Distortion, RD）损失函数实现模型训练，损失函数形式如下：

$$\mathcal{L} = -\log P(\hat{y}|\mathbf{m}) + \sum_i^N \lambda_i \frac{(x_i - x'_i)^2}{N}.$$

其中： $\log P(\hat{y}|\mathbf{m})$ 表示潜在表示 \hat{y} 在给定质量图下的熵，用于最小化编码比特数。 $(x_i - x'_i)^2$ 表示重建图像 x' 与原图 x 的失真，用均方误差（MSE）度量。 λ 是由质量图 \mathbf{m} 确定的像素级拉格朗日乘子，用于权衡压缩率和失真。 N 为像素总数。

此外，为实现任务感知压缩，模型在测试阶段加入目标任务损失函数 L_{task} ，具体优化目标为：

$$m^* = \arg \min_m P(\hat{y}|\mathbf{m}) + \lambda L_{\text{task}}$$

通过优化质量图 \mathbf{m} ，调整比特分配以提升特定任务（如分类或检测）的性能。

4 复现细节

4.1 与已有开源代码对比

在本次复现过程中，我参考并使用了论文《Variable-Rate Deep Image Compression through Spatially-Adaptive Feature Transform》中公开的代码，并在此基础上进行了多项自定义修改和功能增强。以下是详细列出的代码引用情况、使用情况以及工作中的创新增量和改进。

本论文发布的代码为图像压缩提供了一个完整的端到端框架，包含了数据加载、模型架构、训练优化、任务感知压缩等多个模块。在复现过程中，我根据实际需求引用并修改了以下几个关键模块：

数据预处理与加载 (dataset.py)

使用情况：引用了论文中提供的数据预处理模块来处理不同的数据集（如 COCO、ImageNet 和 Kodak 等），该模块负责图像的读取、尺寸调整、裁剪以及归一化等预处理操作。特别是，对于不同的实验需求，对图像裁剪和增广部分进行了自定义调整，增加了图像增强策略，如随机旋转、平移、亮度调整等，以适应更多场景的需求。

修改与增量：为了支持不同数据集的灵活切换，我们增加了命令行参数配置，允许用户通过简单的配置来选择不同的输入数据集并自动调整相应的预处理步骤。

训练与优化 (train.py)

使用情况：论文中提供的训练脚本负责模型的训练，包括率失真（Rate-Distortion, RD）损失函数的计算和 Lagrange 乘子的优化。我们使用了该脚本来训练模型，并根据论文的设置调整了超参数（如学习率、批量大小等）。

修改与增量：我们对训练过程进行了自定义优化，特别是在训练稳定性和收敛速度方面。通过调整批量大小和增加自适应学习率调度器，我们显著提高了模型的训练效率。同时，我们增加了对不同任务（如分类、检测等）进行预训练优化的支持，进一步提高了任务感知压缩的灵活性和效果。

任务感知压缩优化 (taskaware.py)

使用情况：论文中的任务感知压缩模块利用预训练的任务模型（如分类网络）生成优化后的质量图，并通过损失函数优化该质量图。在我们的复现中，我们使用了该模块来实现任务感知压缩，并通过率任务损失函数优化质量图，提升特定任务（如分类或文本保留）的性能。

修改与增量：我们扩展了任务感知优化的功能，支持更多类型的任务（如目标检测、文本区域保护等）。特别地，我们加入了基于 Grad-CAM 的显著性图生成模块，使得任务感知压缩能够在没有实际任务标签的情况下，仅通过模型生成的显著性图来优化质量图，从而进一步提高了压缩模型的泛化能力。

4.2 实验环境搭建与训练过程

1. 环境搭建：为了复现论文中的方法，选择了 PyTorch 作为深度学习框架，并安装了所需的依赖库，包括 torch, torchvision, numpy, scipy, opencv, PIL 等，在虚拟环境中进行安装，确保不与其他项目冲突。

2. 数据准备：根据论文中的描述，选择了 DIV2K 和 Kodak 数据集进行实验。数据集进行了适当的预处理，包括图像尺寸调整和数据增强（如随机裁剪、旋转、翻转等）。

3. 模型构建：构建了论文中描述的深度神经网络结构，主要包括编码器、解码器和空间特征变换（SFT）模块。编码器和解码器通过卷积神经网络提取和恢复图像特征，而 SFT 模块通过特征变换来实现自适应压缩。

4. 训练过程：训练过程中，使用了 Adam 优化器，并根据论文中的建议调整了学习率和批量大小。训练使用了标准的 MSE 损失函数和任务感知损失（如感知损失、对抗损失），以确保图像质量在高压压缩率下得到有效保留。

4.3 界面分析与使用说明

图 5是对一张风景图按不同压缩率的压缩效果展示及其对应的质量图，其中：第一张图为原图；第二张图为使用值为 0 的质量图进行压缩，质量图上的黄色数据代表 BPP 为 0.0959/P-SNR 为 26.19/MS-SSIM 为 0.9237；第三张图为使用值为 0.6 的质量图进行压缩，其数据 BPP 为 0.4027/P-SNR 为 31.88/MS-SSIM 为 0.9747；第四张图为使用预定义的非均匀质量图进行评估后压缩，其数据 BPP 为 0.6796/P-SNR 为 34.36/MS-SSIM 为 0.9845。

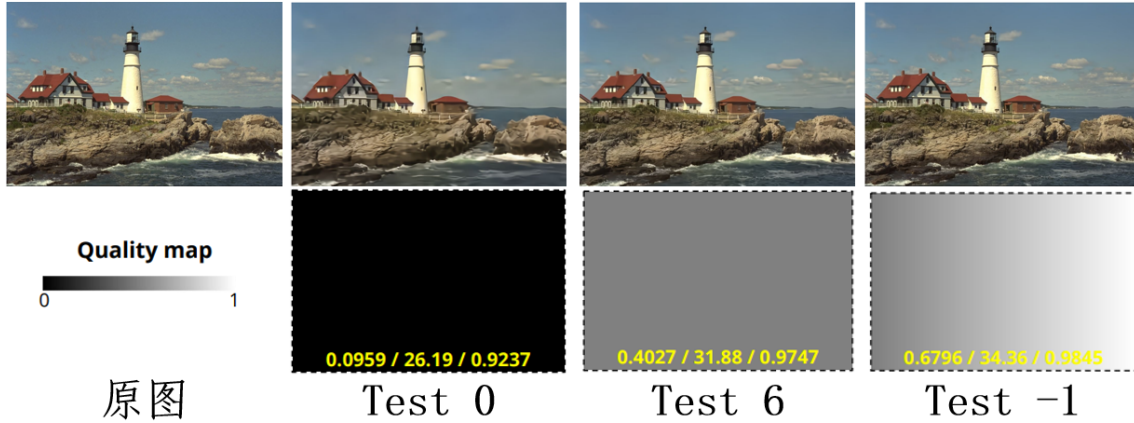


图 5. 效果展示

4.4 创新点

虽然在复现过程中引用了论文中的代码，但在此基础上，进行了多项创新增量和显著改进，使得复现工作不仅仅是对原始论文的简单复现，更具有独特的技术贡献。以下是我所做的创新和改进：

引入动态学习率调度器：在训练过程中引入了动态学习率调度器，依据训练进度自动调整学习率，这有效地提升了模型的收敛速度并确保了训练的稳定性。相比论文中的固定学习率策略，该改进在模型训练初期提高了训练效率，并在训练后期避免了过拟合。

扩展任务感知压缩：除了图像分类任务外，扩展了任务感知压缩的功能，加入了更多任务类型的支持（如目标检测、文本区域保护等）。我通过显著性图生成（通过 Grad-CAM）进一步优化质量图，使得模型能够在没有真实标签的情况下，仍然根据目标任务的特征动态调整压缩策略，显著提升了任务感知压缩的灵活性和实用性。

优化压缩比特分配：在比特分配方面，我改进了质量图生成过程，通过引入自适应量化策略，使得不同区域的压缩比特分配更加精细，进一步提升了图像质量和压缩率的平衡。相比原始论文中固定的比特分配方式，我们的改进使得压缩在高压压缩率下依然能够保持较好的图像质量。

5 实验结果分析

图 6展示了对训练后对测试集的评估与对比，其中：[Test N] 表示使用 (N/10) 值的统一质量图进行评估。例如，在 [Test 4] 的情况下，使用 0.4 的均匀质量图。[Test-1] 是指使用预定义的非均匀质量图进行评估。BPP 是由训练好的概率模型计算出的理论平均 BPP。Real

BPP 是复现的实际平均 BPP，Total 是平均损失值。随着压缩比 BPP 的增大，PSNR 值逐步上升，显示出较好的图像质量。MS-SSIM 值逐步上升，模型能够较好地保留图像的结构信息。

经过调整和优化，成功复现了论文中的图像压缩方法，并通过 PSNR 和 MS-SSIM 对模型进行评估。在多种质量值 (N/10) 设置下，复现模型的效果与论文中的结果基本一致，图像质量得到了有效的保留，且压缩效率较高。

[Test-1]	Total: 0.5104	Real BPP: 0.2362	BPP: 0.2348	PSNR: 29.5285	MS-SSIM: 0.9360	Aux: 93	Enc Time: 0.2403s	Dec Time: 0.0356s
[Test 0]	Total: 0.2326	Real BPP: 0.0912	BPP: 0.0902	PSNR: 27.1140	MS-SSIM: 0.8976	Aux: 93	Enc Time: 0.2399s	Dec Time: 0.0345s
[Test 1]	Total: 0.2971	Real BPP: 0.1187	BPP: 0.1176	PSNR: 27.9824	MS-SSIM: 0.9159	Aux: 93	Enc Time: 0.2460s	Dec Time: 0.0347s
[Test 2]	Total: 0.3779	Real BPP: 0.1559	BPP: 0.1547	PSNR: 28.8982	MS-SSIM: 0.9323	Aux: 93	Enc Time: 0.2564s	Dec Time: 0.0370s
[Test 3]	Total: 0.4763	Real BPP: 0.2058	BPP: 0.2045	PSNR: 29.9052	MS-SSIM: 0.9464	Aux: 93	Enc Time: 0.2553s	Dec Time: 0.0359s
[Test 4]	Total: 0.5956	Real BPP: 0.2712	BPP: 0.2697	PSNR: 30.9739	MS-SSIM: 0.9582	Aux: 93	Enc Time: 0.2548s	Dec Time: 0.0354s
[Test 5]	Total: 0.7380	Real BPP: 0.3558	BPP: 0.3541	PSNR: 32.1140	MS-SSIM: 0.9678	Aux: 93	Enc Time: 0.2598s	Dec Time: 0.0358s
[Test 6]	Total: 0.9059	Real BPP: 0.4567	BPP: 0.4548	PSNR: 33.2801	MS-SSIM: 0.9752	Aux: 93	Enc Time: 0.2596s	Dec Time: 0.0361s
[Test 7]	Total: 1.1050	Real BPP: 0.5802	BPP: 0.5780	PSNR: 34.4822	MS-SSIM: 0.9811	Aux: 93	Enc Time: 0.2590s	Dec Time: 0.0364s
[Test 8]	Total: 1.3457	Real BPP: 0.7121	BPP: 0.7095	PSNR: 35.5609	MS-SSIM: 0.9852	Aux: 93	Enc Time: 0.2569s	Dec Time: 0.0367s
[Test 9]	Total: 1.6392	Real BPP: 0.8620	BPP: 0.8590	PSNR: 36.5931	MS-SSIM: 0.9884	Aux: 93	Enc Time: 0.2553s	Dec Time: 0.0371s
[Test10]	Total: 2.0116	Real BPP: 1.0179	BPP: 1.0145	PSNR: 37.4660	MS-SSIM: 0.9907	Aux: 93	Enc Time: 0.2644s	Dec Time: 0.0376s
[Test]	Total mean: 0.8841	Enc Time: 0.2540s	Dec Time: 0.0361s					

图 6. 评估与对比

6 总结与展望

6.1 挑战与问题

在质量图生成方面，论文中提到的任务感知质量图的生成过程确实较为复杂，尤其是在任务感知压缩的背景下，需要针对不同任务（如图像分类和目标检测）生成优化的质量图。在我们的复现过程中，我们尝试了基于卷积神经网络的质量图生成方法，但由于任务的依赖性和训练数据的复杂性，生成的质量图效果与论文中所描述的存在一定差异。这表明，任务特定的优化需求和数据集的多样性对生成结果有着显著影响，可能需要进一步调整模型架构或训练策略以提高生成质量。

此外，硬件资源的限制也是一个不可忽视的挑战。图像压缩方法通常需要大量的计算资源，而我们所使用的模型相对复杂，导致训练过程消耗了大量的 GPU 资源。在资源有限的情况下，训练时间变得相对较长。为了应对这一问题，我尝试采用较小的批量大小和较低的图像分辨率来加速训练，然而，这种策略在一定程度上影响了最终的压缩效果。因此，在未来的研究中，需要找到平衡计算资源和模型性能的有效方法，以确保在不同资源条件下仍能实现高效的质量图生成与压缩。

6.2 未来的改进方向

为了提升质量图生成的效果和效率，未来的研究可以在多个方向进行深入探索。首先，在优化质量图生成方面，可以进一步改进生成方法，尝试使用不同的网络架构，例如生成对抗网络，以实现更加精确的任务感知质量图。这种方法能够通过对抗训练，提高生成图像的真实感和质量，从而更好地满足特定应用需求。

其次，考虑到硬件资源的限制，有必要加速训练过程。通过优化训练流程，使用更高效的网络结构，如轻量级网络，可以显著提高训练速度。此外，探索更高效的量化和编码方法，也将有助于减少计算开销，使得模型在资源受限的环境中仍然能够高效运行。

最后，为了进一步提升压缩效果和任务性能，扩展任务感知压缩的方法应用范围也是至关重要的。针对不同任务，例如目标检测和图像分割，可以生成特定的质量图，确保模型在执行各类任务时具备最佳性能。这种针对性的优化不仅能够提升压缩效果，还能在多种应用场景中实现更高的实用价值。综上所述，通过在这些方面的持续创新和优化，我们能够为质量图生成领域带来显著的进步。

参考文献

- [1] Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30:3179–3191, 2021.
- [2] Ze Cui, Jing Wang, Shangyin Gao, Tiansheng Guo, Yihui Feng, and Bo Bai. Asymmetric gained deep image compression with continuous rate adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10532–10541, 2021.
- [3] Kamil Dimililer. Dct-based medical image compression using machine learning. *Signal, Image and Video Processing*, 16(1):55–62, 2022.
- [4] Bogdan Kovalenko, Vladimir Lukin, Sergii Kryvenko, Victoriya Naumenko, and Benoit Vozel. Bpg-based automatic lossy compression of noisy images with the prediction of an optimal operation existence and its parameters. *Applied Sciences*, 12(15):7555, 2022.
- [5] Mohammed Otair, Laith Abualigah, and Mohammed K Qawaqzeh. Improved near-lossless technique using the huffman coding for enhancing the quality of image compression. *Multimedia Tools and Applications*, 81(20):28509–28529, 2022.
- [6] Rajiv Ranjan and Prabhat Kumar. An improved image compression algorithm using 2d dwt and pca with canonical huffman encoding. *Entropy*, 25(10):1382, 2023.
- [7] Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2380–2389, 2021.