**中南民族大学**

**毕业论文(设计)**

**学院： 计算机科学学院**

**专业： 自动化类 年级： 2019级**

**题目： 基于深度学习的图像上色系统设计**

**学生姓名：胡小浪 学号：201921099119**

**指导教师： 姚为 职称: 副教授**

**2023年** 5 **月**

**中南民族大学本科毕业论文（设计）原创性声明**

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。本人完全意识到本声明的法律后果由本人承担。

作者签名： 年 月 日

目录

[摘要： 1](#_Toc135002778)

[关键词： 1](#_Toc135002779)

[Abstract: 1](#_Toc135002780)

[Key words 1](#_Toc135002781)

[1 绪论 2](#_Toc135002782)

[1.1 图像上色问题的数学本质 2](#_Toc135002783)

[1.2 图像上色方法 3](#_Toc135002784)

[1.2.1 早期方法 3](#_Toc135002785)

[1.2.2 基于样例的方法 3](#_Toc135002786)

[1.2.3 基于深度学习的方法 4](#_Toc135002787)

[1.3 研究意义和应用 4](#_Toc135002788)

[1.4 本文的论文结构 5](#_Toc135002789)

[2 相关理论方法与技术 6](#_Toc135002790)

[2.1 计算机视觉领域的主要研究问题 6](#_Toc135002791)

[2.2 卷积神经网络 6](#_Toc135002792)

[2.3 预训练模型 8](#_Toc135002793)

[2.4 生成对抗网络 9](#_Toc135002794)

[2.5 基于Python 的深度学习框架 11](#_Toc135002795)

[3 基于预训练模型和生成对抗网络的图像上色方法 12](#_Toc135002796)

[3.1 总体结构 12](#_Toc135002797)

[3.2 详细步骤 13](#_Toc135002798)

[4 实验与结果分析 16](#_Toc135002799)

[4.1 数据集与实验设计 16](#_Toc135002800)

[4.2 基于基础模型的图像上色 18](#_Toc135002801)

[4.3 基于生成对抗网络与预训练模型的图像上色 19](#_Toc135002802)

[4.4 GAN模型的优化实验 21](#_Toc135002803)

[4.5 实验总结 22](#_Toc135002804)

[4.6 展望 24](#_Toc135002805)

[致 谢 25](#_Toc135002806)

[参考文献 25](#_Toc135002807)

基于神经网络的图像上色系统设计

摘要：本文研究了基于卷积神经网络（CNN）和生成对抗网络(GAN)的图像上色方法。本文首先介绍了这两种方法的基本原理和实现细节，然后分析了它们在不同评价指标上的性能，并在实际应用场景中进行对比。通过实验结果对比，发现使用基于生成对抗网络和预训练模型的图像上色方法在能生成更逼真的彩色图像。图像上色问题，即从输入的灰度图像生成与原始彩色图像尽可能接近的彩色图像，具有广泛的应用价值，如老照片修复、黑白电影上色等。

关键词：图像上色, 生成对抗网络, 预训练模型, 卷积神经网络, 深度学习

System design for image coloring based on neural networks

**Abstract:** Adversarial Networks (GANs) and pre-trained models, as well as those based on Convolutional Neural Networks (CNNs). The image colorization problem, which involves generating a color image as close as possible to the original color image from an input grayscale image, has broad application value, such as old photo restoration and black-and-white film colorization. We first introduce the basic principles and implementation details of these two methods and then analyze their performance on various evaluation metrics and compare them in practical application scenarios. Through the comparison of experimental results, we find that image colorization methods based on GANs and pre-trained models have advantages in image quality and evaluation metrics. However, methods based on CNNs also have certain application value in some scenarios. This study provides useful references for solving the image colorization problem and lays a foundation for research and application in related fields.

**Key words**: Image Colorization; Generative Adversarial Networks; Pre-trained Models; Convolutional Neural Networks; Deep Learning

1. 绪论

随着数字图像处理和计算机视觉技术的迅速发展，图像上色问题已成为一个热门研究领域。图像上色，即根据输入的灰度图像（或黑白图像）生成与原始彩色图像尽可能接近的彩色图像[1]，具有广泛的应用价值，如老照片修复、黑白电影上色、艺术创作等。然而，由于多种颜色可能具有相似的灰度值，图像上色问题具有固有的模糊性。此外，复杂场景和多样性要求解决方案具有较强的泛化能力。因此，开发一种有效的图像上色算法对研究者们具有重要意义。

1. 图像上色问题的数学本质

图像上色问题是计算机视觉和图像处理领域的一个重要研究问题，其核心任务是将给定的灰度图像（黑白图像）通过特定算法或方法赋予丰富的颜色信息，使其转变为具有真实感和视觉吸引力的彩色图像。这一问题在许多实际应用场景中具有重要价值，例如老照片修复、黑白电影上色、动漫创作和艺术作品的数字处理等。

图像上色问题可以分为两种类型：自动上色和半自动上色。自动上色是指系统根据输入的灰度图像，通过学习颜色分布和图像特征等信息，完全自动地为图像上色。这种方法通常需要对算法进行大量的训练，以便系统能够学习到复杂的颜色模式和图像结构，从而在各种场景下都能实现较好的上色效果。

而半自动上色则需要人类用户提供部分颜色信息，例如标注某个区域的颜色或者为特定物体提供参考颜色。系统根据用户提供的颜色信息，结合图像特征和颜色传播算法等技术，为其他未标注的区域进行上色。半自动上色方法可以在一定程度上减轻用户的操作负担，同时充分利用人类对颜色知觉的直观理解，从而实现更符合用户期望的上色效果。

数字图像是由像素组成的二维矩阵，每个像素都包含一个颜色值。在计算机图形学和图像处理中，颜色值的表示方式有很多种，其中最常见的是灰度值和RGB颜色模型。

在灰度图像中，像素的颜色值通常用一个介于0（黑色）和255（白色）之间的整数表示，该整数反映了图像在该像素位置的亮度信息。灰度图像只包含单一通道的信息，因此无法表达丰富的颜色信息。而在彩色图像中，像素的颜色值通常由三个量表示：红色、绿色和蓝色（RGB）。如图1-1所示。

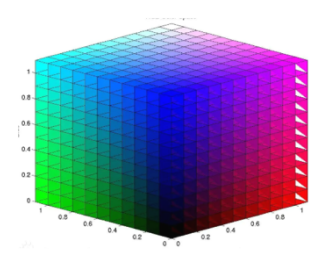


图1-1 RGB颜色空间

每个分量也是一个介于0和255之间的整数，组合起来可以表示大约1,677万种不同的颜色。RGB颜色模型是一种加色模型，其中红、绿、蓝三种基本颜色以不同比例混合，形成各种颜色。例如，RGB值（255,0,0）表示纯红色，而（0,255,0）表示纯绿色。当三个分量的值相同时，它们表示的是灰度值，如（128,128,128）表示中等灰度。

除了RGB颜色模型外，还有其他颜色模型，如HSV（色相、饱和度、明度）和CMYK（青、洋红、黄、黑）等。这些颜色模型在不同的应用场景下具有不同的优势。例如，HSV模型在处理人类对颜色的感知问题时更具优势，而CMYK模型则常用于印刷行业。

图像上色问题的数学本质就在于为灰度图像的每个像素分配一个合适的RGB颜色值，使得整个图像具有自然和真实的颜色效果。

通过解决这个优化问题，能够得到图像的最后着色效果。这种方法让用户有机会通过提供部分颜色信息，对着色结果进行指导和调整。

1. 图像上色方法
2. 早期方法

早期的图像颜色化方法主要依赖于人工规则和启发式方法，这类方法试图从图像的结构属性、纹理细节以及局部关联性等方面获取颜色信息。这些方法一般需要用户提供一定的颜色数据，随后通过特定的算法将这些数据扩展到整个图像上。

例如，Levin等人提出了一个交互式的颜色传播算法，用户通过在图像中的关键位置涂上颜色，然后算法基于图像的梯度和边缘信息自动将颜色扩散到整个图像。这一方法利用了图像中的局部连通性，使得相邻区域具有相似颜色的概率较高。

### 基于样例的方法

基于样例的图像颜色化方法是一种从已知彩色图像中学习颜色变换的技术，将参考图像中的颜色应用到目标灰度图像中。这类方法通常需要一个或多个彩色图像作为参考，然后通过分析参考图像的颜色特征和结构信息，将这些特征映射到目标灰度图像中。这种方法的关键在于找到参考图像与目标图像之间的相似性度量，以便正确地传递颜色信息。

基于样例的方法在某些场景下可以取得较好的颜色效果，尤其是当参考图像与目标图像在结构、纹理和场景上具有较高相似性时[2]。然而，这种方法对参考图像的选择具有较高的依赖性，当参考图像与目标图像的内容差异较大时，颜色化效果可能不尽如人意。此外，这种方法在处理具有复杂纹理和细节丰富的图像时可能面临挑战，因为找到合适的颜色映射关系变得更加困难。

1. 基于深度学习的方法

随着深度学习迅速发展，许多基于深度学习的图像着色方法受到了广泛关注。这些方法通常采用卷积神经网络（CNN）作为基本框架，学习从灰度图像到彩色图像的映射关系。卷积神经网络通过多层卷积和池化操作，能够有效地提取图像的局部特征和全局结构信息。在这方面，Iizuka等人提出了一个端到端的CNN结构，实现了全自动化的图像颜色化。他们的方法包括两个子网络：一个全局网络，用于捕获图像的全局信息，如场景类型和光照条件；一个局部网络，用于提取图像的局部纹理和细节。这两个子网络的输出被融合在一起，以生成彩色图像。该方法在多个数据集上取得了优异的成绩，显示了其在自动图像颜色化方面的强大潜力。

然而，尽管Iizuka等人的方法在很多场景下取得了令人满意的结果，但仍然存在一定的局限性。比如，在处理拥有复杂纹理和细节的图像时，这种方法的性能仍有提升空间。

1. 研究意义和应用

在计算机视觉领域，自动图像着色技术具有显著的价值。它不仅有助于推动图像处理和神经网络技术的创新，而且为相关研究领域带来新的视角和方法。深入研究生成对抗网络、预训练模型训练的模型与CNN卷积神经网络图像上色模型之间的差异，可以帮助我们更好地理解不同神经网络结构和训练策略在图像上色任务中的性能优劣。这将有助于寻找更高效的途径来提高图像着色的质量和效率，同时为其他计算机视觉任务（如图像分割、物体识别和生成任务）提供借鉴。研究图像上色问题还能积累关于神经网络设计和训练的宝贵经验，进一步推动计算机视觉技术的发展。

在实际应用中，自动图像着色技术具有广泛的价值。例如，在媒体行业，它可以用于电影、动画、漫画等领域的黑白影片、动画和漫画的彩色复原，从而降低手工上色的成本和时间。在艺术创作中，艺术家可利用这一技术将手绘草稿或黑白素描迅速转换为彩色作品，提升创作效率和质量。此外，自动图像上色技术还可以应用于文化遗产保护、智能摄影、教育领域、游戏开发、广告设计、社交媒体、无人机与机器人视觉系统以及地理信息系统等多个方面，为各行业带来便利和创新。

在此背景下，本文提出了一种新的图像上色算法，该算法结合了深度学习和先进的图像处理技术，通过对基于生成对抗网络（GAN）和预训练模型的图像上色方法进行研究和分析，本文旨在探索如何提高图像上色的质量和效果。本文的研究目标是设计一个能够处理复杂纹理和细节，同时保持颜色自然、生动的图像上色模型。本文将使用最新的神经网络结构和优化策略，对模型进行训练和调整，以期达到更好的上色效果。本文的研究不仅有望进一步推动图像上色技术的发展，也将为相关领域的研究和应用提供新的思路和工具。在接下来的论文中，将详细介绍本文的研究方法和实验结果，以及本文的模型在各种图像上色任务中的应用效果。

1. 本文的论文结构

为了介绍基于神经网络的图像上色系统设计的内容，本文分了四个部分。

第一部分介绍了图像上色问题的本质，以及图像上色问题的研究意义与应用。并针对图像上色的各种方法进行了叙述。

第二部分介绍了图像上色问题的相关理论方法和技术，指出了CV领域目前面临的主要研究问题，并对卷积神经网络，预训练模型，生成对抗网络（GAN）以及现有的深度学习框架进行了介绍。

第三部分介绍了基于生成对抗网络与预训练模型的图像上色方法的总体结构和编程实现。

第四部分则对本文的实验方案进行了描述，以及实验结果的分析和未来的工作展望。

1. 相关理论方法与技术
2. 计算机视觉领域的主要研究问题

下表总结了计算机视觉领域的四个主要研究方向，以及它们的主要任务、方法和关键技术。

表2-1 研究方向

|  |  |  |
| --- | --- | --- |
| 研究方向 | 主要任务 | 方法与技术 |
| 图像识别 | 从图像中识别并分类物体 | 特征提取、特征描述、分类；卷积神经网络（CNN）等深度学习技术 |
| 目标检测 | 识别图像中的多个物体及定位位置 | 滑动窗口、基于区域提议、单阶段检测方法；R-CNN、YOLO、SSD等深度学习算法 |
| 图像分割 | 将图像划分为具有相似特征的区域 | 阈值分割、边缘检测、区域生长、图割；全卷积网络（FCN）、U-Net、Mask R-CNN等深度学习算法 |
| 图像增强 | 提高图像的视觉质量 | 直方图均衡化、滤波器、变换域方法；卷积神经网络（CNN）、生成对抗网络（GAN）、自编码器（Autoencoder）等深度学习方法 |

图像识别是通过识别并分类图像中的物体，图像识别任务可以为计算机视觉应用提供基础。现代方法通常采用深度学习技术，如卷积神经网络（CNN），以自动提取特征并进行分类。目标检测任务关注于在图像中识别并定位多个物体。深度学习算法如R-CNN、YOLO和SSD已经在目标检测任务中取得了显著的成果，能够有效地处理这类问题。图像分割任务通过将图像划分为具有相似特征的区域，可以为图像理解和高级计算机视觉任务提供更细致的信息。全卷积网络（FCN）、U-Net和Mask R-CNN等深度学习算法在图像分割任务中表现出色。图像增强任务旨在提高图像的视觉质量，使其更易于分析和处理。深度学习方法如卷积神经网络（CNN）、生成对抗网络（GAN）和自编码器（Autoencoder）在图像增强任务中已经取得了显著的成果，能够自动学习复杂的图像特征并生成具有更好视觉质量的图像。

1. 卷积神经网络

卷积神经网络（Convolutional Neural Network，CNN）是一种特殊的深度学习模型，主要用于处理具有网格结构的数据，如图像。卷积神经网络通过卷积操作捕捉局部特征，能够有效处理图像的平移、缩放和旋转等变换。CNN在图像识别、目标检测、图像分割等计算机视觉任务中取得了显著的成果。最基础的CNN结构如下图2-1所示：

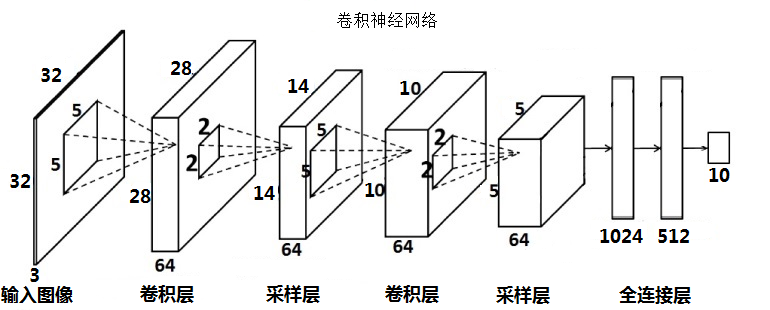


图2-1 卷积神经网络结构图

卷积层（Convolutional Layer）是CNN的核心组件[3]。卷积层使用卷积核（也称为滤波器）在输入图像上执行卷积操作，从而提取图像的局部特征。对图像（不同的数据窗口数据）和滤波矩阵（一组固定的权重：因为每个神经元的多个权重固定，所以又可以看做一个恒定的滤波器filter）做内积（逐个元素相乘再求和）[4]，卷积操作的公式如下：

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>I</mi><mrow><mi>c</mi><mi>o</mi><mi>n</mi><mi>v</mi></mrow></msub><mfenced><mrow><mi>x</mi><mo>,</mo><mo>&#xA0;</mo><mi>y</mi></mrow></mfenced><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><munder><mo>&#x2211;</mo><mi>a</mi></munder><munder><mo>&#x2211;</mo><mi>b</mi></munder><mi>w</mi><mfenced><mrow><mi>a</mi><mo>,</mo><mo>&#xA0;</mo><mi>b</mi></mrow></mfenced><mo>&#xA0;</mo><mo>&#xB7;</mo><mo>&#xA0;</mo><mi>I</mi><mfenced><mrow><mi>x</mi><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>a</mi><mo>,</mo><mo>&#xA0;</mo><mi>y</mi><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>b</mi></mrow></mfenced></mstyle></math>"}

其中，{"mathml":"<mml:math style=\"font-family:stix;font-size:16px;\" xmlns:m=\"http://schemas.openxmlformats.org/officeDocument/2006/math\" xmlns:mml=\"http://www.w3.org/1998/Math/MathML\"><mml:mstyle mathsize=\"16px\"><mml:msub><mml:mi>I</mml:mi><mml:mrow><mml:mi>c</mml:mi><mml:mi>o</mml:mi><mml:mi>n</mml:mi><mml:mi>v</mml:mi></mml:mrow></mml:msub></mml:mstyle></mml:math>"} 表示卷积后的结果，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>w</mi></mstyle></math>"} 表示卷积核，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mo>(</mo><mi>a</mi><mo>,</mo><mo>&#xA0;</mo><mi>b</mi><mo>)</mo></mstyle></math>"} 表示卷积核的位置。

激活函数（Activation Function）为CNN引入非线性，使得网络能够学习到更复杂的特征。表2-2 是一个包含常用激活函数的表格：

表2-2 常用激活函数

|  |  |  |  |
| --- | --- | --- | --- |
| 激活函数 | 符号 | 公式 | 特点 |
| ReLU | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>R</mi><mi>e</mi><mi>L</mi><mi>U</mi><mfenced><mi>x</mi></mfenced></mstyle></math>"} | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>f</mi><mfenced><mi>x</mi></mfenced><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>x</mi><mfenced><mrow><mn>0</mn><mo>,</mo><mo>&#xA0;</mo><mi>x</mi></mrow></mfenced></mstyle></math>"} | 非线性，易于计算，但可能出现"死神经元"现象 |
| Leaky ReLU | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>L</mi><mi>e</mi><mi>a</mi><mi>k</mi><mi>y</mi><mi>R</mi><mi>e</mi><mi>L</mi><mi>U</mi><mfenced><mi>x</mi></mfenced></mstyle></math>"} | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>f</mi><mfenced><mi>x</mi></mfenced><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>x</mi><mfenced><mrow><mi>a</mi><mi>x</mi><mo>,</mo><mo>&#xA0;</mo><mi>x</mi></mrow></mfenced></mstyle></math>"} | ReLU的改进，防止"死神经元"，$\alpha$为一个很小的正数，如0.01 |
| Sigmoid | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>&#x3C3;</mi><mfenced><mi>x</mi></mfenced></mstyle></math>"} | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>f</mi><mfenced><mi>x</mi></mfenced><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mfrac><mn>1</mn><mrow><mn>1</mn><mo>+</mo><msup><mi>e</mi><mrow><mo>-</mo><mi>x</mi></mrow></msup></mrow></mfrac><mspace linebreak=\"newline\"/></mstyle></math>"} | 非线性，平滑，输出范围(0, 1)，但可能出现梯度消失问题 |
| Tanh | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>tan</mi><mi>h</mi><mfenced><mi>x</mi></mfenced></mstyle></math>"} | {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>f</mi><mfenced><mi>x</mi></mfenced><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mfrac><mrow><msup><mi>e</mi><mi>x</mi></msup><mo>-</mo><msup><mi>e</mi><mrow><mo>-</mo><mi>x</mi></mrow></msup></mrow><mrow><msup><mi>e</mi><mi>x</mi></msup><mo>+</mo><msup><mi>e</mi><mrow><mo>-</mo><mi>x</mi></mrow></msup></mrow></mfrac></mstyle></math>"} | 非线性，平滑，输出范围(-1, 1)，但可能出现梯度消失问题 |

池化层（Pooling Layer）用于降低特征图的尺寸，从而减少计算量和参数数量，同时提高模型的泛化能力。常用的池化操作包括最大池化（Max Pooling）和平均池化（Average Pooling）。全连接层（Fully Connected Layer）将卷积和池化层提取到的特征映射最终的输出空间，如分类或回归任务[5]。全连接层通常位于网络的末端，并使用Softmax或Sigmoid等激活函数输出概率分布。损失函数（Loss Function）用于比对模型预测结果与真实结果之间的差异。常用的损失函数包括交叉熵损失（Cross-Entropy Loss）、均方误差损失（Mean Squared Error Loss）等[6]。

CNN的训练通常采用随机梯度下降（Stochastic Gradient Descent，SGD）或其变体（如Adam、RMSProp等）进行优化[7]。通过大量的训练数据，CNN能够自动学习到图像中的有用特征，并在各种计算机视觉任务中取得优异的表现。

1. 预训练模型

预训练模型是在大量数据上预先训练好的神经网络模型。这些模型已经学习到了数据中的一般特征，因此可以直接应用到新任务上，或者通过迁移学习进行微调。使用预训练模型的优势包括：节省训练时间、减少数据需求、提高模型性能。

在计算机视觉领域，预训练模型主要针对图像分类任务。这些模型通常在大规模的数据集（如ImageNet）上进行训练，学习到了丰富的视觉特征，因此可以直接应用到新任务上，或者通过迁移学习进行微调。使用预训练模型可以节省训练时间，因为预训练模型已经在大量数据上进行了训练，因此可以直接用于新任务，无需从头开始训练模型[8]。这大大减少了训练时间，提高了研究效率。还可以减少数据需求。由于预训练模型已经学习到了一般特征，它们可以在相对较小的数据集上进行微调。这意味着即使在数据量有限的情况下，也可以利用预训练模型获得较好的性能。并且能够提高模型性能。预训练模型已经在大规模数据集上学习到了丰富的视觉特征，这有助于提高新任务中模型的性能。通过迁移学习和微调，预训练模型可以在新任务上达到更高的精度和鲁棒性。

总之，在计算机视觉领域，预训练模型已经成为一种非常重要的技术，可以帮助研究人员快速实现高性能的图像处理任务。在本次研究中，预训练模型作为基础，以提升模型的表现，以期在图像分类、目标检测等任务上取得更好的成果。

表2-3展示了计算机视觉领域常见的预训练模型及其特点。

表2-3 常见预训练模型

|  |  |  |
| --- | --- | --- |
| 预训练模型 | 提出者 | 特点 |
| VGG | 牛津大学视觉几何组 | 结构简单，易于理解，参数较多，计算量较大 |
| ResNet | 微软研究院 | 通过残差连接解决梯度消失问题，可以训练非常深的网络 |
| Inception | Google | 采用Inception模块，有效增加网络宽度和深度 |
| MobileNet | Google | 轻量级网络，适用于移动设备和嵌入式应用 |
| EfficientNet | Google | 良好的性能和效率权衡，通过联合缩放实现 |

VGG是由牛津大学视觉几何组（Visual Geometry Group）提出的VGG模型，包括VGG-16和VGG-19两个版本[9]。这两个版本分别包含16层和19层，结构简单，易于理解。但由于参数较多，计算量较大，VGG模型在实际应用中可能会受到计算资源的限制。ResNet是由微软研究院提出的残差网络（Residual Network），通过引入残差连接解决了深度网络中的梯度消失问题，可以训练非常深的网络[10]。常见的ResNet版本包括ResNet-18、ResNet-34、ResNet-50、ResNet-101和ResNet-152。这些模型在图像分类任务上取得了非常好的性能，适用于对性能要求较高的应用场景。Inception是由Google提出的Inception（又称为GoogLeNet）网络，采用Inception模块，可以有效地增加网络的宽度和深度。Inception网络有多个版本，包括Inception-v1、Inception-v2、Inception-v3和Inception-v4。这些模型在多个计算机视觉任务上表现出色，适用于各种场景。MobileNet是由Google提出的轻量级卷积神经网络，专为移动设备和嵌入式应用设计。MobileNet采用了深度可分离卷积（Depthwise Separable Convolution），大幅减少了参数数量和计算量。MobileNet有两个版本：MobileNet-v1和MobileNet-v2。这些模型在实时图像处理任务上具有优势，适用于计算资源受限的场景。EfficientNet是由谷歌提出的一种高效卷积神经网络。EfficientNet通过对网络的深度、宽度和分辨率进行联合缩放，实现了良好的性能和效率权衡。EfficientNet有多个版本，如EfficientNet-B0到EfficientNet-B7。这些模型在多个任务上取得了领先的性能，适用于对性能和效率都有要求的应用场景。在本研究中，采用ResNet预训练模型作为基础，以提升模型的表现。

1. 生成对抗网络

生成对抗网络（Generative Adversarial Network，GAN）是一种深度学习模型，由 Ian Goodfellow 等人于 2014 年提出[11]。GAN 通过同时训练两个神经网络来生成数据：一个是生成器（Generator），另一个是判别器（Discriminator）。生成器负责产生新的数据，而判别器负责判断数据是真实的还是生成器所生成的伪造数据。这两个网络互相竞争，相互各自提高各自的生成能力和判别能力[12]。GAN 在图像生成、文本生成和样本扩增等领域取得了显着的成果。GAN的基本原理图如图2-2。

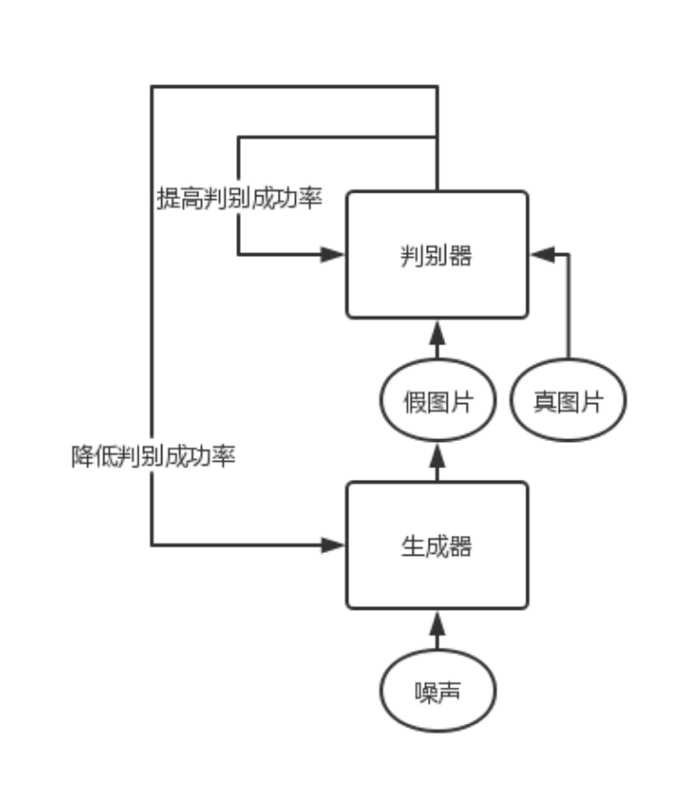
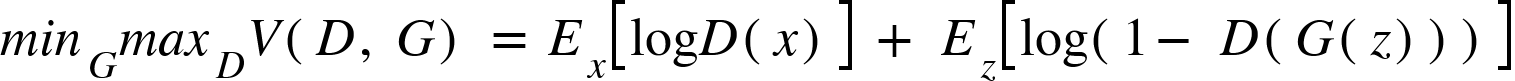


图2-2 GAN基本原理图

生成器（Generator，简称G），是一种神经网络结构，通常采用卷积神经网络（CNN）或循环神经网络（RNN）等模型。生成器的主要任务是从随机噪声中产生与真实数据具有相似特征的伪造数据。在这个过程中，生成器的目标是尽可能地欺骗判别器，使其无法区分生成数据和真实数据之间的差异。判别器（Discriminator，简称D）同样是一种神经网络结构，也通常采用CNN或RNN等模型。判别器的主要职责是对输入数据进行判断，以确定数据是真实数据还是由生成器产生的伪造数据。在这个过程中，判别器的目标是尽可能地识别出生成器生成的数据，从而提高区分真实数据与伪造数据的能力。

训练过程中，生成器和判别器分别更新它们的参数，以优化各自的目标函数。具体而言，生成器试图最大化判别器犯错的概率，而判别器试图最小化犯错的概率。这个过程可以用以下目标函数表示：



其中，wpsoffice 和 wpsoffice 分别表示对真实数据 x 和随机噪声 z 的期望。这个目标函数体现了一个“零和博弈”的思想，即生成器和判别器互相竞争，共同进化。

GAN的训练需要一定的技巧，因为生成器和判别器之间可能出现模式崩溃（Mode Collapse）、梯度消失等问题[13]。为了解决这些问题，研究者提出了许多GAN的变种，如DCGAN（Deep Convolutional GAN）、WGAN（Wasserstein GAN）、CycleGAN等。这些变种对原始GAN的结构和目标函数进行了改进，提高了模型的稳定性和性能。

1. 基于Python 的深度学习框架

在本研究中，我使用了TensorFlow作为深度学习框架。TensorFlow是由Google Brain团队开发的一个开源深度学习框架，广泛应用于实现各种深度学习模型，如卷积神经网络（CNN）、循环神经网络（RNN）和生成对抗网络（GAN）等[14]。它具有灵活、高效、可扩展的特点，支持CPU、GPU以及TPU等多种硬件加速。

我选择 TensorFlow 的原因有以下几点：首先，TensorFlow 2.x 具有更加简洁且易于理解的 API，以及默认启用的 Eager Execution，让开发过程更加直观和高效。其次，TensorFlow 2.x 集成了 Keras，这是一个高级深度学习 API，能让用户以简明的方式定义、训练和评估模型。此外，TensorFlow 社区庞大，拥有丰富的学习资源、技术支持和开源项目，有助于快速解决问题和实现创新。最后，TensorFlow 在多种硬件平台（如 CPU、GPU 和 TPU）上都能实现优秀的性能，且具备强大的可扩展性和部署能力，可以满足各种场景和需求。

1. 基于预训练模型和生成对抗网络的图像上色方法
2. 总体结构

整体结构包括一个生成器（Generator）和一个判别器（Discriminator），它们共同构成了生成对抗网络（GAN）。GAN的目标是训练生成器生成逼真的数据，同时优化判别器以便更好地区分真实数据和生成数据。

生成器采用编码器-解码器（Encoder-Decoder）结构，包含多个卷积层和反卷积层。编码器部分逐步提取输入特征，解码器部分逐步放大尺寸以生成输出。在编码器和解码器之间，本文使用跳跃连接将编码器的特征与解码器的特征连接，以提高网络性能并减少信息损失。激活函数（如LeakyReLU和tanh）被用于引入非线性。部分结构如图3-1。

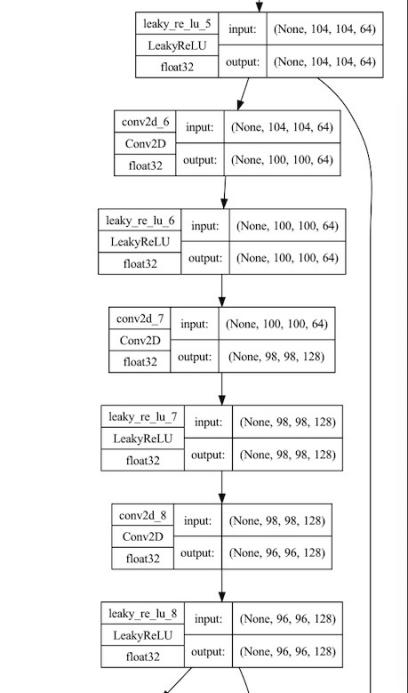


图3-1 生成器部分结构

判别器是一个卷积神经网络，由多个卷积层组成，逐渐减小输入尺寸并提取特征。判别器的输出层使用sigmoid激活函数，将结果映射到[1]区间，表示输入数据是真实数据的概率，判别器部分结构如图3-2。

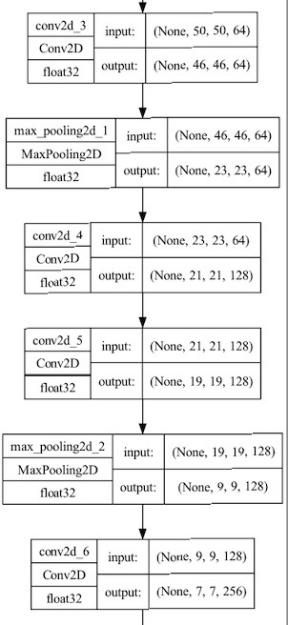


图3-2 判别器部分结构

在训练过程中，生成器和判别器进行对抗式学习。生成器试图生成逼真的数据以欺骗判别器，而判别器努力区分真实数据和生成数据[15]。这个过程通过最小化生成器损失和判别器损失来实现。在训练结束后，生成器可用于生成新的逼真数据，而判别器可以用于评估￼数据的真实性。

1. 详细步骤

该方法详细的训练步骤如下图3-3:

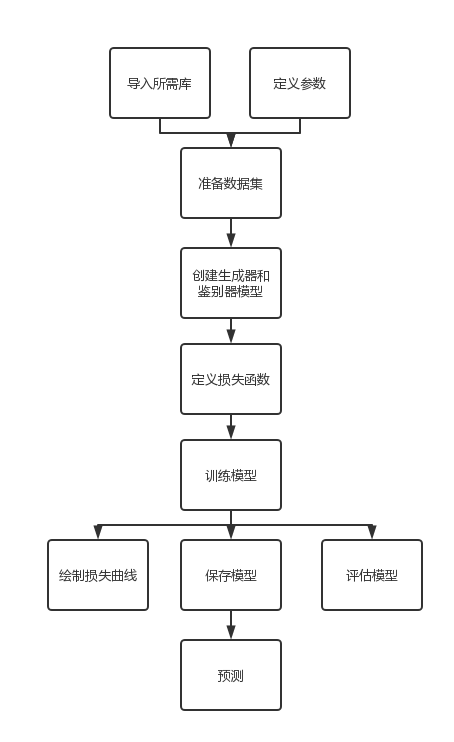


图3-3 模型训练步骤

为了实现生成器和判别器的训练过程，首先需要导入一些库，具体导入库以及版本号如表3-1。

表3-1

|  |  |  |
| --- | --- | --- |
| 库名 | 版本 | 功能描述 |
| os | - | 用于文件和目录操作 |
| cv2 | 4.x | 用于图像处理和计算机视觉任务 |
| numpy | 1.x | 用于数组操作和数学计算 |
| tensorflow | 2.x | 用于构建、训练和部署深度学习模型 |
| PIL | 8.x | 用于图像处理，如打开、保存和显示图像 |
| sklearn | 0.24.x | 用于数据预处理、模型评估和性能指标 |

接着设置训练参数，例如epoch数量、批处理大小和图像大小。之后，从主目录中读取数据集，将彩色图像转换为灰度图像，并将数据集分为训练集、测试集和验证集。

生成器负责将灰度图像转换为彩色图像，本文使用卷积层、激活函数和卷积转置层构建生成器。判别器负责判断生成的彩色图像是否真实，本文使用卷积层、激活函数、最大池化层和全连接层构建判别器。

然后定义生成器和判别器的损失函数，并使用Adam优化器分别优化生成器和判别器的权重[16]。训练过程中，使用TensorFlow的GradientTape记录生成器和鉴别器的梯度，然后应用优化器更新权重。遍历训练集，对每个批次执行训练步骤，并在每个epoch结束时计算验证损失。

在训练过程中，将生成器损失、鉴别器损失和验证损失绘制成图形，以便观察训练过程。训练完成后，将训练好的生成器和鉴别器模型保存到磁盘。评估模型时，计算模型在验证集上的平均损失。最后，使用生成器对测试集进行预测，并将生成的彩色图像与原始图像和灰度图像进行比较，以评估模型的表现。

1. 实验与结果分析
2. 数据集与实验设计

我从现有的公开数据集中筛选了2500张图片作为我的数据集（部分图片如下图所示），为了提高模型的泛化能力和鲁棒性，我还利用 TensorFlow、Numpy 及 os 库对数据集图片进行预处理，其中包括图像裁剪、图像增 强和图像格式转换等操作，实现去除非目标区域的干扰、 增强图像特征、降低图像维度等效果，在一定程度上提高了神经网络图像分类的准确率[17]。

数据集被划分为训练集、验证集和测试集，其中训练集用来训练模型，验证集用来调整模型参数，测试集用来评估模型性能(图像上色)，划分比例8:1:1。

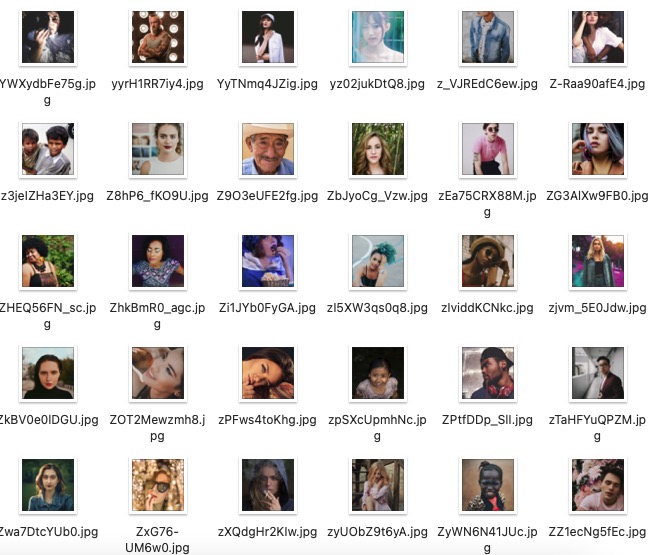


图4-1数据集部分图像

实验设计分为三个阶段，旨在分别验证CNN基础模型的可行性，比较基于GAN和预训练模型的性能，优化GAN模型，流程步骤如图4-2。

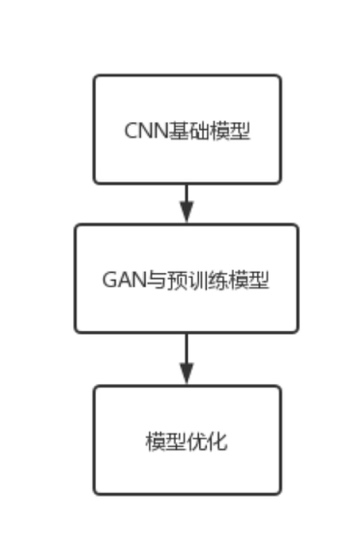


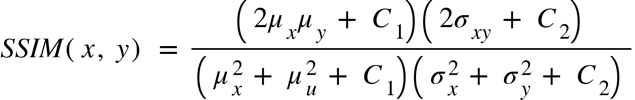
图4-2 实验步骤

第一阶段：为了验证卷积神经网络（CNN）在图像上色任务中的可行性，我构建了一个中等规模的CNN分类器模型，并把2500张图片按照75%，15%，15%的比例分为训练集、验证集和测试集。模型主要由卷积层、激活函数和池化层组成。实验所使用的处理器为3.6 GHz 四核Intel Core i3, 内存 16 GB 2667 MHz DDR4，显卡为AMD Radeon RX 460/560 2 GB，512G的固态硬盘。编程环境使用的是Python3.10，tensorflow-2.12.0 版本。操作系统选用的macOS Monterey。

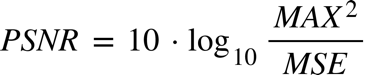
第二阶段：使用构建好的基于GAN和预训练模型的模型，在同样的数据集上进行训练，验证和测试。由于该模型复杂度较高，使用训练基础模型的训练条件已经难以完成任务，所以我将实验条件更换为：3.6 GHz 四核Intel Core i5, 内存 32 GB 2667 MHz DDR4，显卡为Geforce RTX 3060 8GB，512G的固态硬盘。编程环境使用的是Python3.10，tensorflow-2.12.0 版本。操作系统选用的windows 10。

第三阶段：对训练好的GAN模型进行图像上色，并根据最终上色效果，对模型的参数进行调整，包括网络结构，学习率，迭代次数，批次处理大小等。

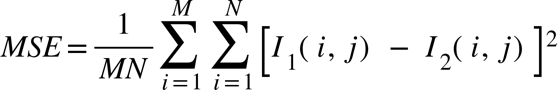
在以上各个阶段，本文对不同模型在测试集上的性能分别进行评估。使用的评价指标包括峰值信噪比（PSNR）、结构相似性指数（SSIM）。这两个指标的定义如下：



其中，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>"}和{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>y</mi></mstyle></math>"}分别表示两个图像，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>&#x3BC;</mi><mi>x</mi></msub></mstyle></math>"}和{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>&#x3BC;</mi><mi>y</mi></msub></mstyle></math>"}分别表示它们的均值，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msubsup><mi>&#x3C3;</mi><mi>x</mi><mn>2</mn></msubsup></mstyle></math>"}和{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msubsup><mi>&#x3C3;</mi><mi>y</mi><mn>2</mn></msubsup></mstyle></math>"}分别表示它们的方差，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>&#x3C3;</mi><mrow><mi>x</mi><mi>y</mi></mrow></msub></mstyle></math>"}表示它们的协方差，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>C</mi><mn>1</mn></msub></mstyle></math>"}和{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>C</mi><mn>2</mn></msub></mstyle></math>"}是常数，用于避免分母为零的情况。



其中，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>M</mi><mi>A</mi><mi>X</mi></mstyle></math>"}表示图像可能的最大像素值（例如，8位图像的最大像素值为255），wpsoffice表示均方误差，定义为：



这里，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>I</mi><mn>2</mn></msub></mstyle></math>"}和{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>I</mi><mn>2</mn></msub></mstyle></math>"}表示两个图像，{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>M</mi></mstyle></math>"}和{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>N</mi></mstyle></math>"}分别表示图像的行数和列数。结构相似性值为0～1，数值越大表示失真越小，即模型的上色效果越好[18]；同样峰值信噪比越 大，表示上色图像越接近真实图像[19]。

1. 基于基础模型的图像上色

下图是模型的损失率，本文可以看到，随着迭代次数的增加，模型的损失率在逐渐下降，这表明模型的学习能力在不断地提高。

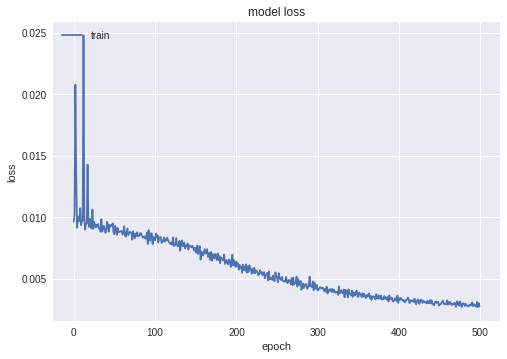


图4-3 CNN模型训练的损失率Loss曲线

下面是模型的精度曲线图，本文可以看到，随着迭代次数的增加，模型的精度在逐渐升高，这表明模型正在逐渐提高对训练数据的拟合能力。

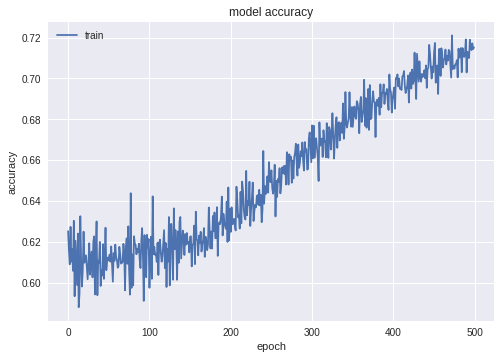


图4-4 CNN模型训练的精度曲线

下图是使用训练好的模型上色的效果图， 其中左边是原本的彩色图片，中间是灰度图片，右边是上色效果图，图片标题显示了当前图片的峰值信噪比（PSNR）和结构相似性指数（SSIM）。

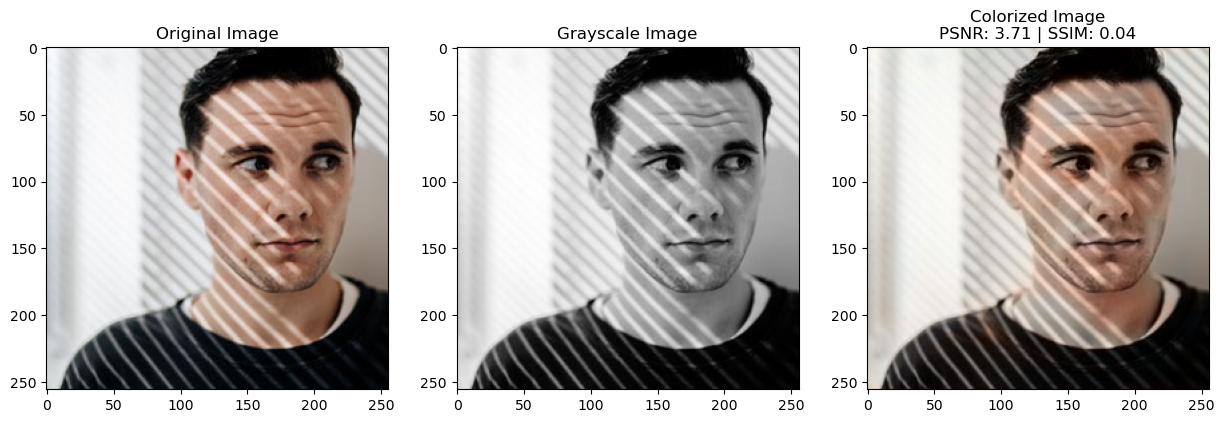
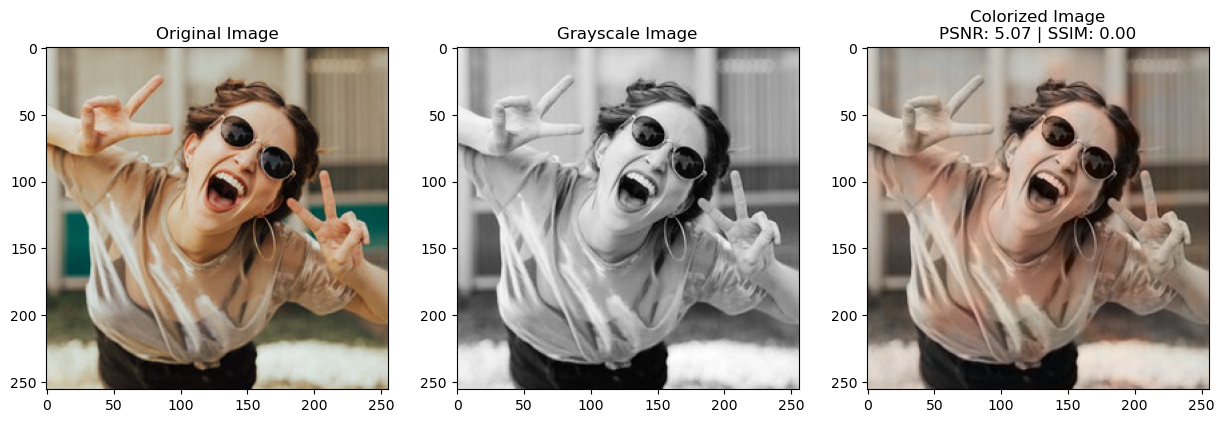
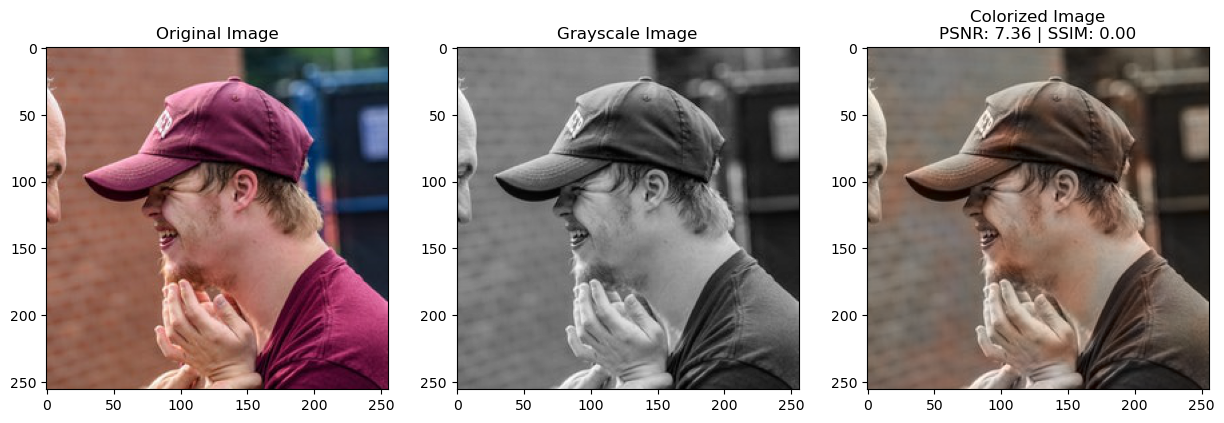
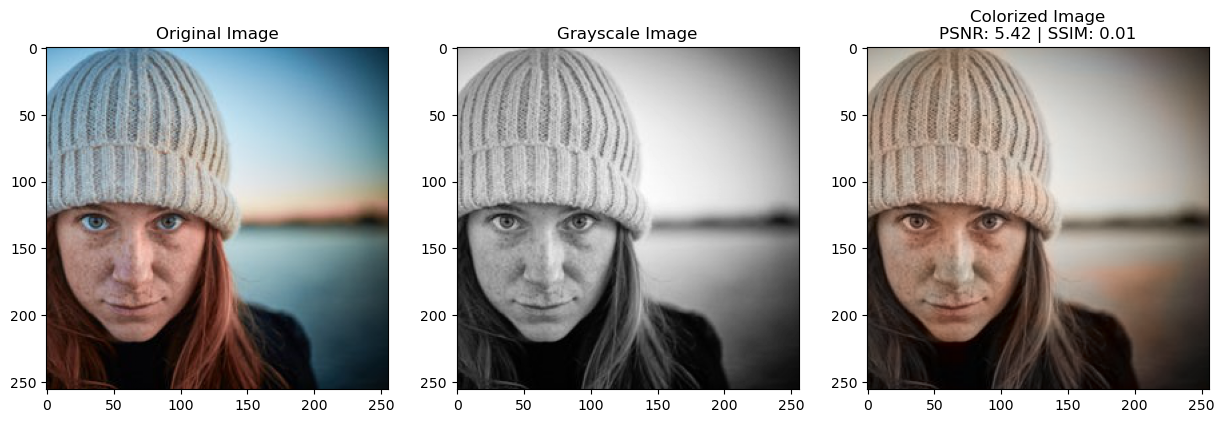


图4-5 CNN模型上色效果图

1. 基于生成对抗网络与预训练模型的图像上色

图4-6是使用训练好的模型上色的效果图，通过肉眼观察，可以很明显的看出上色后的图像，颜色相对CNN上色的图像更丰富，更逼真。

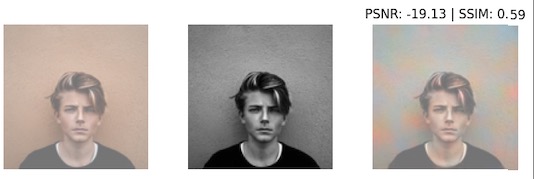
图4-6 GAN模型上色效果图

图4-7是模型训练过程中的生成器、判别器与验证损失的变化曲线图。观察曲线，我们不难发现，虽然图像色彩更加逼真了，但生成器的损失变化幅度几乎没有，而判别器的损失在整体降低的趋势下还是有有大的波动，这里可能的情况是判别器过于强大，导致生成器无法有效的学习，也有可能是学习率设置的不合理。

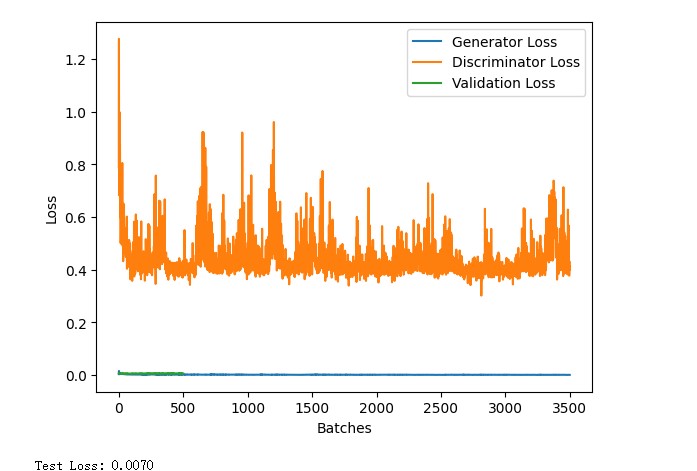


图4-7 GAN模型训练过程损失曲线

由于硬件的限制，加上GAN的训练是一个复杂且具有挑战性的任务，可能需要进行大量的实验和调整，本文就不做更多关于判别器结构的调整。而对于学习率的优化，本文将会引入一个新的策略，来使得上色后的图像更加逼真。

1. GAN模型的优化实验

一个设计良好的学习率策略可以显著提高模型的收敛速度和收敛精度，减少模型的训练时间[16]。学习率衰减是一种在训练过程中调整学习率的策略。初始阶段，大的学习率可以快速地进行权重更新，随着训练的进行，通过逐渐降低学习率，可以更好地将模型参数优化到更精细的状态。为了测试该模型是否还能进一步优化，生成更真实的图像，我尝试对模型添加了学习率衰减策略。

考虑到硬件条件的限制，在本文的模型中选择了多项式衰减策略[20]。训练初期的学习率设定为0.005，这是本文开始训练时的学习率，较大的学习率有利于我们快速找到一个相对优秀的解。衰减步数，本文设定为10000。这个参数决定了学习率减少到最终学习率所需要的步骤数量。最终的学习率设定为0.0001，这是我们训练结束时的学习率，较小的学习率可以帮助模型在训练的后期阶段进行更精细的调整。多项式的幂设定为1.0，这个参数决定了学习率衰减的速率，幂越大，学习率衰减得越快。通过合理地设定的衰减步数和多项式的次数参数，多项式衰减策略可以在训练初期帮助模型快速地找到一个合适的解，而在训练后期帮助模型更精细地优化这个解。

优化后的模型训练效果如图4-8，和上一步实验中的图4-6相对比，颜色相对来说更加丰富了，图像也更加逼真了，这说明设置一个合理的学习率对于模型的训练是有很大帮助的。采用学习率衰减策略，可以避免了我们做大量重复的训练，让我们可以快速的找到适合的解。

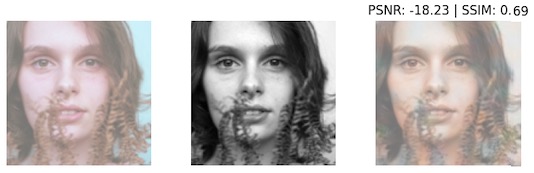


图4-8 GAN模型上色效果图

图4-9 是优化后的损失值变化曲线图，和未做优化的模型得到的变化曲线相比，他们具有相同的特征，这也更加说明了本文中所编写的判别器结构过于强大而导致生成器不能非常有效的学习。

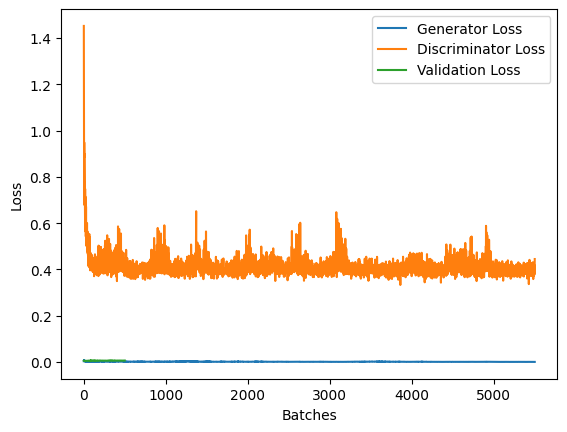


图4-9 优化后的GAN模型训练过程损失曲线

1. 实验总结

在本文的实验中，我们对比了基于生成对抗网络（GAN）和预训练模型的图像上色模型，以及基于卷积神经网络（CNN）的图像上色模型。对比最终图像上色的结果，从客观层面上我发现：

基于生成对抗网络（GAN）和预训练模型的图像上色模型在保留图像细节和生成自然色彩方面具有较好的性能，这表明生成器的预训练模型能够提取更丰富的图像特征，这说明生成器与判别器之间的对抗训练更有助于生成逼真的彩色图像。

根据实验中所使用的评价指标峰值信噪比（PSNR）和结构相似性指数（SSIM），我们可以得出以下结论：从图像质量角度来看，生成对抗网络（GAN）和预训练模型的图片着色方法通常展现出更优秀的性能。这说明这种方法在生成自然色彩和保留图像细节方面都具有较强的能力。而在训练效率和模型复杂度方面，基于卷积神经网络（CNN）的图像上色模型具有优势，这代表了这种方法更适合在计算资源有限的设备上进行部署和使用。

两种方法各有优劣，可分别用于不同的场景。基于生成对抗网络（GAN）和预训练模型的图像上色模型在图像质量方面表现较好，更适用于对图像质量要求较高的场景，如媒体制作、艺术创作等。而基于卷积神经网络（CNN）的图像上色模型在训练效率和模型复杂度方面具有优势，更适合在计算资源有限的设备上进行部署和应用，如智能手机、相机等。

1. 展望

这次实验使我更深入地了解了这两种图像上色方法的优缺点。以下是一些实验的延伸方向和未来工作：针对生成对抗网络（GAN）和预训练模型的图像上色方法，可以进一步优化和调整生成器和判别器的结构，让他们进行更有效的对抗，避免出现一方过于强大的情况，以改善生成图像的质量。本文中生成器和判别器的结构并不是此次实验的最优解，但是也取得了不错的效果，而结构的调整多种多样，这也更加印证了基于生成对抗网络和预训练模型的上色相对基础CNN是能够取得更好的效果的。

不同的优化器也有可能会影响模型的训练效果，所以还可以选择使用更合适的优化器。同时，还可以尝试将其他类型的生成对抗网络（如CycleGAN、StarGAN等）应用于图像上色任务，以提高模型的泛化能力。而对于基于CNN的图像上色方法，可以尝试使用更深层次的网络结构，以提取更丰富的图像特征。此外，可以尝试将注意力机制、残差连接等先进技术融入到模型中，以改善上色效果。

在本次实验中，虽然上色的图像逐渐变得更加逼真，但我们没有保证每一次实验使用相同的验证集和测试集，这是因为实验中的验证集和测试集都是按照比例随机挑选出来的，这是不足之处，可以为未来的探索中加以改进。还可以尝试使用更多样化的数据集，以测试模型在不同领域和场景下的性能。此外，可以研究如何利用无监督学习或半监督学习方法，以减少对大量标注数据的依赖。

为了提高图像上色系统的实用性，可以考虑开发一款用户友好的软件或移动应用程序，将上色功能与其他图像处理功能（如滤镜、美颜等）结合起来，为用户提供一站式的图像编辑解决方案。还可以进一步探讨将图像上色技术应用于其他相关领域，如视频上色、虚拟现实、增强现实等，为这些领域的发展提供技术支持。

通过深入研究这些方向，我们可以在计算机视觉领域取得更多突破，拓宽图像上色技术的应用范围，为实际应用场景提供更优质的解决方案。

# **致 谢**

在这里，我要向姚为老师表达我最真挚的感激之情。在整个大学阶段，姚为老师用他严谨的教学精神和生动有趣的授课方式给我留下了深刻的印象。在完成毕业设计和撰写论文的过程中，姚老师提供了许多宝贵的启示和建议，并多次悉心指导和修改。我由衷感谢姚老师的专注、幽默和亲和力，以及与我们进行的诸多学术交流。

同时，我也要向本专业的其他杰出教师表示感激，他们分别是张志俊老师、樊金荣老师、陈勉老师、苗祖英老师、周熠老师、张俊敏老师、吴桂华老师、唐玉玲老师、魏红昀老师、程立老师、朱宗晓老师、王黎老师、田微老师、王超老师以及李薇老师等。在我大学四年的求学过程中，他们给予了我专业知识上的指导和支持。

我还要感谢我的父母，感谢他们在过去的岁月里对我的学业给予了支持和鼓舞。我特别要感谢我的女朋友卢燕飞，她的理解和支持让我有更多的精力专注于学业，她的鼓励和爱护让我在遇到困难和挑战时充满了勇气。她的存在是我大学生活中最重要的力量源泉。

同时感谢学校营造的宁静、协调的学习氛围。最后，我要对我那些亲密的室友们表达感激，在四年的大学生活里，我们互相扶持，共同进步。

# 参考文献

[1] 苏梦晶. 基于生成式对抗网络的表情合成算法研究 [D]; 贵州大学, 2022.

[2] 王鑫鑫. 基于图像解纠缠的上色方法研究 [D]; 大连理工大学, 2020.

[3] 王庆, 姚俊, 谭文禄. 基于Faster R-CNN的排水管道缺陷检测研究 [J]. 软件导刊, 2019, 18(10): 40-4,9.

[4] 韩旭. 彝文古籍字符检测和识别的研究与实现 [D]; 西南大学, 2020.

[5] 杜振国. 卷积神经网络模型压缩方法研究与应用 [D]; 南昌大学, 2021.

[6] 李林, 黄志华, 张晶福. 基于深度学习的SAR目标识别技术研究 [J]. 长江信息通信, 2021, 34(7): 6-10.

[7] 尹文召. 基于深度学习的混合推荐模型研究 [D]; 河北大学, 2020.

[8] 刘志宇, 黄亦翔. 基于深度学习和迁移学习的液压泵健康评估方法 [J]. 机械与电子, 2018, 36(09): 67-71.

[9] 高明旭, 李靖, 朱绪平. 深度学习方法研究综述 [J]. 中国科技信息, 2019, (10): 56-7.

[10] 杨晶东, 杨鑫, 赵诚. 自适应融合残差网在图像分类中应用研究 [J]. 小型微型计算机系统, 2020, 41(02): 399-405.

[11] 王颢. 深度学习在图像识别中的研究与应用 [J]. 科技视界, 2020, (24): 37-8.

[12] 王德文, 李业东. 基于WGAN图片去模糊的绝缘子目标检测 [J]. 电力自动化设备, 2020, 40(5): 188-94.

[13] 吴蔓. 基于生成对抗网络的文本到图像生成算法研究 [D]; 电子科技大学, 2021.

[14] 赛迪智库. 人工智能发展需警惕深度学习框架 [J]. 软件和集成电路, 2018, (2): 92-4.

[15] 刘杰, 王昌达, 武秋敏. 改进的生成对抗网络在轴承故障诊断中的应用 [J]. 噪声与振动控制, 2021, 41(02): 89-92+124.

[16] 刘帆, 刘鹏远, 张峻宁. 一种改进的深度学习模型自适应学习率策略 [J]. 兵工自动化, 2019, 38(01): 72-7.

[17] 赵浩. 基于TensorFlow的卷积神经网络图像分类实践策略研究 [J]. 价值工程, 2020, 39(9): 205-7.

[18] chaiky. 图像质量评价指标 [EB/OL].CSDN, 2022, https://blog.csdn.net/qq\_39751352/article/details/126098081.

[19] 肖海霞, 张峰, 王亚强. 基于生成对抗网络和卫星数据的云图临近预报 [J]. 应用气象学报, 2023, 34(2): 220-33.

[20] 荔七加er. 学习率衰减策略 [EB/OL].知乎, 2022, https://zhuanlan.zhihu.com/p/475824165.