

班 级 _____

学 号 _____

西安电子科技大学

本科毕业设计论文



题 目 基于 GAN 网络的图像恢复方法设计

与实现

学 院 _____

专 业 _____

学生姓名 _____

导师姓名 _____

西安电子科技大学

毕业设计（论文）诚信声明书

本人声明：本人所提交的毕业论文《基于 GAN 网络的图像恢复方法设计与实现》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的无论以何种方式发布的文字、研究成果，均在论文中加以说明；有关教师、同学和其他人员对本文的写作、修订提出过并为我在论文中加以采纳的意见、建议，均已在我的致谢辞中加以说明并深致谢意。

本论文和资料若有不实之处，本人承担一切相关责任。

论文作者：_____

日期_____

摘要

计算机数字图像处理就是把图像的信号通过一定设备转换成数字信号后用计算机进行处理的整个过程。图像复原是计算机数字图像处理领域的一个重要的问题，它的目的是尽可能地减少甚至除去在成像和传输过程中发生的图像质量下降即图像退化。

经典的图像复原方法有维纳滤波，盲反卷积，超分辨率重建、卷积神经网络等等这些模型。这些方法都有各自的缺陷，高维方程求解、需要构造出特定的退化模型，鲁棒性差等诸多问题的存在，大大降低了单一方法在图像复原中的应用范围。而生成对抗网络以其强大的对抗学习能力、自适应性、强鲁棒性以及适合大规模并行处理的优势，被广发应用到各领域。

本文从一般生成对抗网络的基本概念出发，深入研究生成对抗网络理论，旨在传统的生成对抗网络结构基础上改进其内部结构，并基于此理论算法进一步开展图像复原的深入研究。

首先，介绍了图像复原技术的研究背景和研究意义。然后介绍了图像复原的一般退化模型，并且详细阐述了朴素生成对抗网络的相关模型，提出基于生成对抗网络进行图像复原的相关模型设计和改进。基于生成式对抗网络进行图像复原的核心设计，是把进行图像复原的生成器和判断图像质量的判别器都分别用深度卷积神经网络结构来实现，利用生成器和判别器之间的二元零和博弈，使二者分别达到最优，从而使整个生成对抗网络模型达到最优。接着，本文详细阐述了如何具体构建卷积生成对抗网络模型，网络层数、相关参数以及优化函数的选择。最后，论文对整个模型进行了训练测试，构造了对 Comics 数据集进行图像复原的结果并对结果进行了分析。

实验结果表明，基于 GAN 的改进模型 DCGAN 模型进行图像复原，能够较快地收敛，当图像信息损失量低于百分之 50 时都可以很好地复原出原始的图像。

关键词： 图像复原, 生成对抗网络, 卷积神经网络, 零和博弈

ABSTRACT

Computer digital image processing is the entire process of converting an image signal into a digital signal through a certain device and processing it with a computer. Image restoration is an important issue in the field of computer digital image processing. Its purpose is to reduce or even eliminate image quality degradation or image degradation that occurs during imaging and transmission as much as possible.

The classic image restoration methods include Wiener filtering, blind deconvolution, super-resolution reconstruction, convolutional neural network and so on. Each of these methods has its own drawbacks. The existence of many problems such as the solution of high-dimensional equations, the need to construct a specific degradation model, poor robustness, etc., greatly reduce the application scope of a single method in image restoration.

The advantages of Generative Adversarial Nets with its strong countermeasures, adaptability, robustness, and suitability for large-scale parallel processing are widely applied to various fields.

This paper starts from the basic concept of Generative Adversarial Nets, deeply studies the theory of Generative Adversarial Nets, aims to improve the internal structure of the traditional network structure based on the Generative Adversarial Nets, and further studies the restoration of images based on this theoretical algorithm.

First, the paper introduces the research background and research significance of image restoration technology. Then the general degradation model of image restoration is introduced, and the relevant models of naive Generative Adversarial Nets are elaborated. The related model design and improvement of image restoration based on the Generative Adversarial Nets are proposed. The core design of image restoration based on the Generative Adversarial Nets is that both the generator for image restoration and the discriminator for judging image quality are implemented using a deep volumetric network structure, respectively, and the binary zero between the generator and the discriminator is utilized. And the game makes the two achieve optimal respectively, so that the entire Generative Adversarial Nets model is optimized. Then, this paper elaborates on how to build a specific build of the warp machine against the network model, the number of network layers, related parameters, and optimization functions. Finally, the paper conducts

training tests on the entire model, constructs the image restoration results for the Comics dataset, and analyzes the results.

The experimental results show that the image restoration based on the generative countermeasure network model can converge quickly, and the original image can be well restored when the image information loss is less than 50 percent.

Keywords: image restoration, generatrix-resistant network, Convolutional Neural Network, zero-sum game

目录

摘要	i
ABSTRACT	iii
目录	v
第一章 绪论	1
1.1 课题研究背景以及意义	1
1.1.1 研究背景	1
1.1.2 研究意义	1
1.2 研究现状	2
1.3 论文完成的工作	3
1.4 论文内容安排	4
1.5 本章小结	4
第二章 图像复原相关模型介绍	5
2.1 一般退化模型	5
2.2 卷积神经网络	5
2.2.1 概述	5
2.2.2 名词介绍	5
2.2.3 基本的网络结构	6
2.3 差异	6
2.4 本章小结	7
第三章 生成对抗式网络	9
3.1 概述	9
3.2 核心思想	10
3.3 GAN 基本结构	10
3.3.1 生成模型	10
3.3.2 判别模型	11
3.3.3 损失函数	11
3.4 标准模型	12
3.4.1 极小极大博弈	12
3.5 基本工作流程	13
3.6 本章小结	14

第四章 基于 GAN 图像复原的模型设计	15
4.1 朴素 GAN 缺陷	15
4.2 生成网络设计	16
4.3 判别网络设计	16
4.4 损失函数	17
4.5 参数设置	17
4.5.1 生成网络参数设置	17
4.5.2 判别网络参数设置	21
4.6 算法流程	24
4.7 本章小结	24
第五章 实验及结果分析	25
5.1 实验环境	25
5.2 数据准备	25
5.3 结果分析	25
5.3.1 高斯模糊样本集	25
5.3.2 遮蔽模糊样本集 1	26
5.3.3 遮蔽模糊样本集 2	27
5.4 优化训练速度	29
5.5 本章小结	30
第六章 总结和展望	31
6.1 总结	31
6.2 展望	31
致谢	33
参考文献	35

第一章 绪论

1.1 课题研究背景以及意义

1.1.1 研究背景

计算机图像处理起源于 1960 年。伴随着第一台能进行图像处理的巨型计算机的出现,人们开始尝试用计算机进行图形和图像处理。早期图像处理致力于改善图像的质量,评判标准是人的视觉感受。通常输入低质量的图像,通过计算机处理之后输出较高质量的图像。随着计算机图像处理技术在世界范围内的普及和发展,计算机图像处理已经成为工业界需求最大的技术,伴随而来的是人们对计算机图像处理技术的要求越来越高。

现代的图像处理通常指对图像进行分析、加工和处理,使其满足视觉、心理乃至机器人视觉辅助系统要求的技术。图像处理是信号处理在图像领域上的一个应用。目前大多数的图像均是以数字形式存储,因而图像处理很多情况下指数字图像处理。

图像复原作为计算机图像处理的一个重要的分支,在计算机科学和工程领域都有广泛的应用和发展前景。科学界和工业界对计算机图像复原的要求也越来越高,图像复原一直是一个难以完美解决的难点问题。早期的图像复原算法通常是建立图像逆退化模型,把模糊的图像经过逆退化网络,逐步复原出原始图像。因为造成图像失真的噪音千变万化,所以模型的鲁棒性往往较差,且效果不是很理想,只能应对很简单的复原问题。近年来,随着硬件条件的提升以及海量图像数据量的出现,使运用深度神经网络进行图像复原成为了可能,基于此进行图像处理的研究如雨后春笋。2014 年伊恩·古德费洛等人提出了生成对抗网络 (Generative Adversarial Network) 简称 GAN 模型,将机器学习在图像处理领域的应用提上了新的台阶。

1.1.2 研究意义

在人们的日常生活以及工业生产中,图片的产生量和使用量越来越大,因此对图像的质量要求也越来越高。现阶段人工智能技术的一个难点就是解决图像信息丢失的问题,图像复原技术已经成为人工智能技术的一个瓶颈,因此图像复原技术的提高迫在眉睫。

在图像的生成和传输过程中,外界各种因素的干扰,都会使图像出现不同程度的信息损失,这种现象称为图像的退化。成像过程中的退化原因主要集中在相

机的硬件缺陷，相机与物体之间的相对运动，以及相机成像系统的散焦问题等等；传输过程中的退化原因集中在压缩解压缩造成的信息损失。

退化的图像丢失或者异化了很多原始信息，使得获取到的图像信息与真实的图像信息之间存在一定的偏差。在成像和传输系统都难以优化的情况下，想要尽可能的复原出原始图像，这是就需要使用图像复原技术来进行修复。图像复原的目标是降低噪声并且恢复分辨率的损失。

图像复原其实就是图像的退化过程的反过程。利用退化的先验知识，建立相应的数学模型，根据相应的数学模型进行逆推导，用来恢复出最为接近原始图像的数据。这其中，建立相对应的逆数学模型就成为最核心的任务。引起图像退化的原因多种多样，因此要建立十分精确的逆退化模型是十分困难的。早期的图像复原受限于计算机性能较低和图片数据量较少，进行逆退化的数学模型通常比较简单，复原出来的效果也不是太符合人的期望。随着计算机性能的提高，特别是高性能 GPU 的出现，使建立更为复杂精准的逆退化网络模型以及寻找大规模模型参数成为了可能，为图像复原技术的突破提供了新的解决思路。生成对抗网络作为人类计算机网络结构最新的研究成果，在图像一般处理领域表现卓越。研究用新型生成对抗网络来进行图像复原，不仅对计算机图像处理技术的发展有着重要的意义，而且对于人工智能技术瓶颈的突破也提供了全新的解决方案，是人类计算机技术发展的必然结果。

1.2 研究现状

图像复原的研究主要集中在以下几个方面：逆退化数学模型建立，图像复原的具体算法和复原出的图像质量评价体系。不同的问题规模，退化图像的成像模型以及评价标准都会衍生成不同图像复原算法。

现行常用的图像复原算法有如下几种：

- 维纳滤波
- 图像的盲反卷积
- 超分辨率重建
- 卷积神经网络

维纳滤波是 19 世纪 40 年代美国科学家维纳提出的一项重要研究成果。它是一种以最小平方差为最优准则的线性滤波器，因此又叫最小方差滤波，是图像复原中的一种常用手段，主要用来处理先验信息比较丰富的复原问题。维纳滤波核心是利用了平稳随机过程的相关特性和图像频谱特性对退化的图像进行滤波。它的目标是找到一个原始图像的估计，使他们之间的均方差最小。

图像盲反卷积法主要用来处理缺少先验基础信息的复原问题，也是图像复原的一种重要手段。在缺少甚至没有先验信息的情况下，直接依据退化后的模糊图像来估计退化函数，并运用估计的退化函数，构造逆退化模型来恢复出原始的图像信息。图像盲反卷积主要有以下几种算法：露西-理查德森算法，Sato 法，Bussgan 法等其中露西-理查德森算法是一种非线性算法，使用非线性迭代，在计算量，性能方面逗比维纳滤波有一定的提升。

超分辨率技术主要应用在由一个或者多个退化的低分辨率图像的基础之上，逐渐重建出高分辨率原始图像的过程。在日常和工业监控设备，医学成像，遥感卫星图片等领域都有广泛的应用。其中根据根据单个图像重建出高分辨率原始图像叫做单图像分辨率重建。根据多个退化图像重建出高分辨率原始图像的技术，称之为多图像的超分辨率重建。单图像的超分辨率重建是利用部分先验知识构造出高分辨率的原始图像信息。依据先验知识的类别，获取的图像以及使用的方法可以把重建的方法划分为基于正则化的重建方法，基于学习重建方法以及基于插值的重建方法。多图像的超分辨率重建则利用了多个图像之间相互的信息补充，通过在更高的分辨率下的上层网络进行像素点的扩充和融合，从而得到了信息更全，分辨率更高更清晰的图像。

卷积神经网络主要应用在有海量图像数据的基础之上，通过大量的图像退化信息，迭代训练网络的参数。在机器人视觉，和无人驾驶等方向运用广泛，在注重图片整体信息复原的同时，能够注意到局部图像信息的复原。缺点是需要大量调参，大量的样本数据，同时物理含义不够明确，人们很难知道每一个卷积层到底提取到了哪些图片的特征。

在这些方法中，前三种都建立在一般退化模型的基础之上，最后一种则是建立在深度学习模型的基础之上。

1.3 论文完成的工作

本文综述了基于 GAN 改进后的模型，对图像进行复原的一系列原理和实现。具体内容如下：

1. 介绍了图像复原的研究背景和发展现状：相关图像复原技术。
2. 介绍了了有关图像复原的一般退化模型和卷积神经网络模型。
3. 研究了 GAN 的数学原理和具体的网络架构；。
4. 重点阐述了基于 GAN 进行图像复原的模型设计，参数设置以及算法流程。
5. 用 Python 和 tensorflow 实现了基于 GAN 改进后的图像复原模型，对实验结果进行分析并且提出和实现了两种优化训练速度的方法。

1.4 论文内容安排

本论文主要研究了基于 GAN 进行图像复原的数学原理和实现，重点研究了使用开源工具 tensorflow 在 python 环境下实现 GAN 生成对抗式网络的搭建，以及利用此模型进行图像复原。接下来的具体安排如下：

第一章：介绍了图像复原的研究背景以及发展现状。第二章：介绍了图像复原相关模型，一般退化模型和卷积神经网络模型；第三章：介绍了 GAN 网络的基本思想和数学模型；第四章：实现基于 GAN 进行图像复原的模型设计，并且详细给出相关参数设置和算法步骤第五章：进行实验，并对实验结果进行详细的分析第六章：总结和展望。

1.5 本章小结

本章主要介绍了本文的研究背景和意义、研究现状、论文完成的工作以及论文的内容安排。在第一部分，我们简单介绍了计算数字机图像处理技术发展的历程，并指出图像复原是计算机数字图像处理领域的一个重点和难点。第二部分介绍了现在常用的图像复原方法：基于一般退化模型的维纳滤波、盲反卷积、超分辨率重建以及基于机器学习的卷积神经网络。第三部分介绍了本文完成的工作。最后一部分则给出了本文的内容以及章节安排。

第二章 图像复原相关模型介绍

2.1 一般退化模型

图像的退化模型可以用图 2.1 所示的线性不变模型描述：

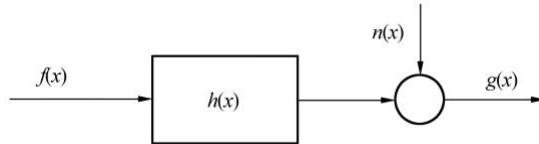


图 2.1 一般图像退化模型

对应的数学表达式为：

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \quad (x, y) \in \Omega \quad (2-1)$$

其中 $\Omega \subset R^2$ 是图像的像素点的全集, $f(x, y)$ 表示原图像, $h(x, y)$ 表示退化模型, $g(x, y)$ 表示退化后的图像, $n(x, y)$ 表示随机噪声, $*$ 代表卷积运算。 $f(x, y)$ 经过一个退化系统 h 的作用后在可加性随机噪声 $n(x, y)$ 的作用下, 退化成图像 $g(x, y)$ 。图像复原的目的就是在 $h(x, y)$ 已知或者未知的情况下, 获得关于图像的近似估计 $f'(x, y)$ 。现实中, 退化因素多种多样, 传统的滤波方法很难得到准确的退化函数。

2.2 卷积神经网络

2.2.1 概述

卷积神经网络 (Convolutional Neural Network) 简称 CNN, 是一种前馈型神经网络, 因为在图像处理中的表现优异, 被计算机视觉领域广泛应用。卷积神经网络相对于一般的深度网络结构, 需要的参数比较少, 训练速度快。在图像分类、图像识别、视频分析、图像复原等领域都有广泛应用。

2.2.2 名词介绍

卷积神经网络中存在这三个基本的数学概念：局部像素感受区间, 权值共享, 池化。

1. 局部像素感受区间。卷积神经网络把单个隐藏节点连接到图像的某个局部区域, 以减少所需要的训练参数, 从而避免了参数爆炸现象的产生。
2. 权值共享。卷积神经网络区别于一般的神经网络, 在同一层中的每一个神经元对应的权值都是相同的, 进一步降低了训练需要的参数个数。

3. 池化。通常情况下, 原始图像输入规格巨大, 但是原图的信息量不必要全部使用, 获取图像的有效信息即可。在实际处理中, 为了加速处理过程, 采用了类似图像压缩的思想, 对图像卷积之后, 通过池化操作来调整图像的输入大小。

2.2.3 基本的网络结构

图 2.2 展示了一个四层的卷积神经网络: 卷积神经网络包括卷积层, 池化层

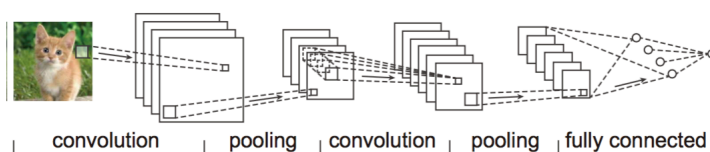


图 2.2 卷积神经网络示意图

(即下采样层) 以及全连接层, 其中除了全连接层只有一层以外, 其他各层都可以有更多层结构。卷积神经网络相比于普通的全连接网络有以下几个优点:

1. 可以适应图像的变化性。图像拥有自己的局部不变特征, 充分利用了图像像素与像素之间的相对位置关系。图像的任意放大、缩小、旋转等都不会影响到视觉系统对于图片局部的识别。
2. 计算量较少。因为卷积神经网络的权值共享、局部像素感受区间以及池化层的存在, 大大减少了训练过程中的参数量, 加快了训练速度并且提升了模型的鲁棒性。对于图像处理相关任务, 卷积神经网络保留了特征参数, 以及减少了不必要的参数, 提升了训练速度, 并且保证了训练质量。

模糊的图像经过卷积神经网络最终映射到原图像, 从而实现了图像复原的效果。通常会采用随机梯度下降和前向传播算法进行网络 (映射) 参数的训练。

2.3 差异

基于图像一般退化模型的图像复原过程, 通常要建立逆退化模型, 明确退化函数, 进而对图像进行逆退化即图像复原, 整处理过程是人为可以预知和感受到的。基于卷积神经网络的图像复原则不需要十分准确地建立逆退化模型, 也不需要明确求解退化函数。它更像是一种图像翻译的过程, 把退化后的图像通过每一层网络映射, 最终映射到原图像。往往只根据输出输出的图像与原图像之间的差距, 运用前向传播算法, 逐层调整网络的参数即可。整个处理过程不为人所感知, 可解释性差, 类似于“黑箱”。

2.4 本章小结

本章主要介绍了基于一般退化模型进行图像复原的数学原理、基于卷积网络进行图像复原的网络结构以及两者的差异。第一部分重点介绍了图像的退化过程，并且推导了用图像的逆退化模型进行图像复原的数学原理。第二部分重点阐述了卷积神经网络的结构和优势，以及进行图像复原的数学原理。最后一部分则总结了两种模型的差异。

第三章 生成对抗式网络

3.1 概述

生成对抗网络 (Generative Adversarial Nets) 简称 GAN, 是一种包含无监督学习领域的人工智能算法, 通过无监督的方式对真实样本的学习, 模拟其数据分布的状况, 以产生相似的样本数据为目的。其网络结构如图 3.1: GAN 由生成网络 G

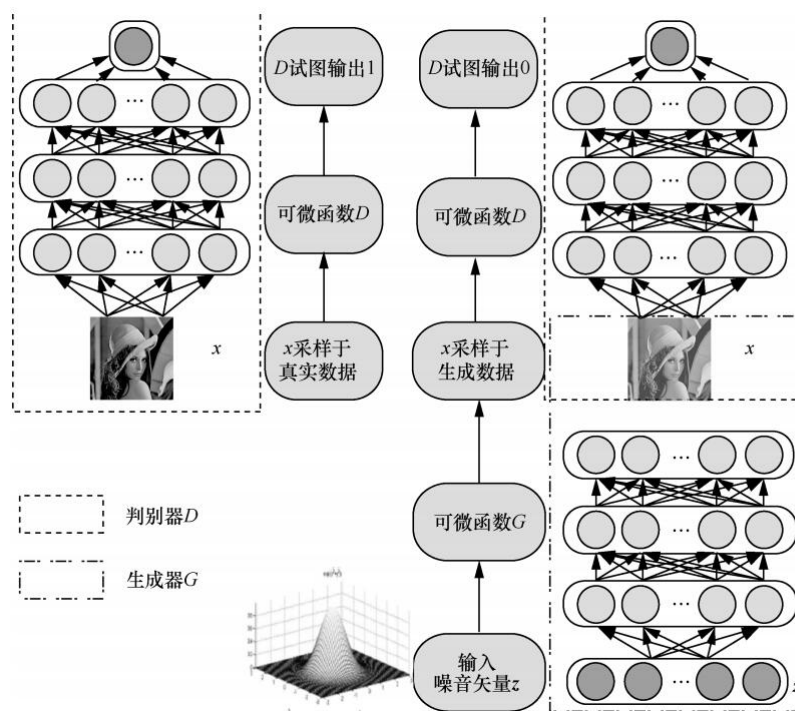


图 3.1 GAN 示意图

和判别网络 D 两个子模型组成, 生成模型的目的是是生成的新样本与真实样本尽可能相似, 而判别模型的目的则是准确无误地区分真实样本和生成样本。在 GAN 提出之前, 生成模型的主要思想是: 模拟真实的样本概率分布。在获取真实样本的概率分布前提下, 通过对其随机采样生成新的样本。鉴于获取真实样本的概率分布难度较大, GAN 提出一种新的思想: 通过学习一组随机变量到真实样本的映射关系, 进而获取一个由多层神经网络组成的模型。GAN 不直接估计真实样本的概率分布, 而是通过模型学习的方式生成与真实样本具有相同概率分布的新样本。生成算法学习类的分布, 判别算法学习类的边界。生成网络的输入是特定函数生成的随机噪声 z , 输出是计算机自己生成的伪图片; 判别网络的输入是 `image`, 包括真实图片 x 以及伪图片 $G(z)$, 输出是一个标量, 用来表示 `image` 是真实图片的概率。判别网络认为 `image` 是真实的图片, 输出 1, 判别网络认为 `image` 是伪图片,

输出 0。生成网络和判别网络，单独迭代交替训练，不断优化，当判别网络无法正确判断数据的来源时，就认为生成网络捕捉到了真实的样本分布。

3.2 核心思想

GAN 衍生于二元零和博弈，其背后的思想是从训练库里获取很多训练样本，从而学习这些训练案例生成的概率。GAN 包含了一对相互对抗的网络模型：生成网络（Generator Network）和判别网络（Discriminator Network）。生成网络不断捕捉训练库里的真实图片的概率分布，加上随机的随机噪声（Random Noise）转变成新的样本。判别网络同时观察真实和造假的数据，判断这个数据是真实的还是伪造生成的。在博弈中为了提高各自的准确率，生成网络和判别网络会不断地提高自己的生成能力和判别能力，优化的最终结果是找到生成网络和判别网络之间的纳什均衡状态。

3.3 GAN 基本结构

3.3.1 生成模型

基于简单的神经网络作为生成模型的结构如图 3.2 所示：

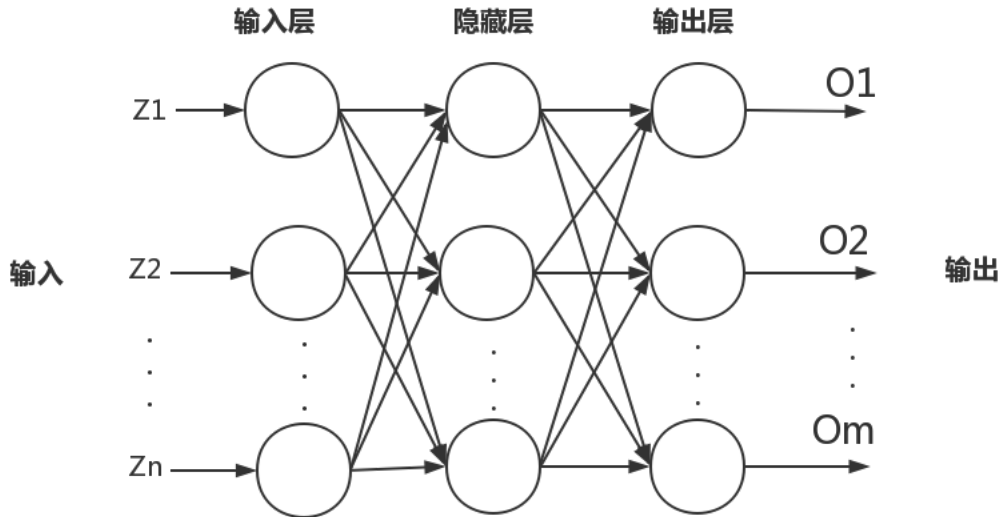


图 3.2 简单神经网络的生成模型

生成模型也是由输入层、隐藏层、输出层组成的三层神经网络结构，不同的是，生成模型神经网络输入的是 n 维服从某一已知概率分布的随机噪声；输出为生成样本。其中 G_L 表示生成网络的层数， G_n^l 表示第 l 层神经元的数量， $G_σ_l(\cdot)$ 表示第 l 层神经元的激活函数， $G_W^l \in R^{n^{l-1} \times n^l}$ 表示第 $l-1$ 层到第 l 层神经元的

权值矩阵, $G_b^l \in R^{n^l}$ 表示第 l 层神经元的偏置量, G^{n^l} 表示第 l 层神经元的输入, $G_a^l \in R^{n^l}$ 表示第 l 层的输出。

3.3.2 判别模型

基于简单的神经网络作为判别模型的结构如图 3.3 所示:

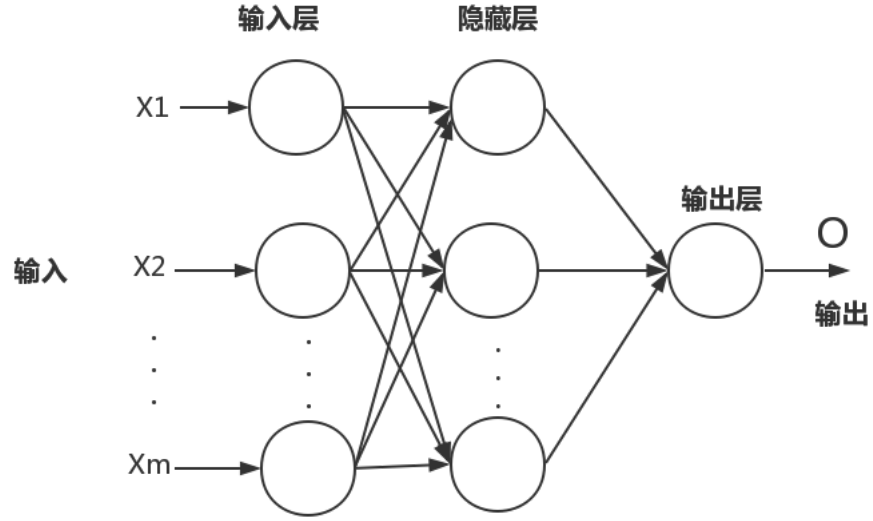


图 3.3 简单神经网络的判别模型

判别模型是由输入层、隐藏层、输出层组成的三层神经网络。该神经网络输入是真实样本或者生成样本, 输出是当前样本为真实样本而非生成样本的概率。 D_L 表示神经网络的层数, D_n^l 表示第 $l-1$ 层到第 l 层神经元的数量, $D_σ_l(\cdot)$ 表示第 l 层神经元对应的激活函数, $D_W^l \in R^{n^{l-1} \times n^l}$ 表示第 $l-1$ 层到第 l 层神经元的权值矩阵, $D_b^l \in R^{n^l}$ 表示第 l 层神经元的偏置量, R^{n^l} 表示第 l 层神经元的输入, $D_a^l \in R^{n^l}$ 表示第 l 层的输出。

3.3.3 损失函数

判别模型和生成模型都有其各自的损失函数。判别模型的末表是准确地将输入的真实样本标记为真, 输入的生成样本标记为假。因此, 在判别模型存在两种损失: 将输入的真实样本标记为假以及将输入的生成样本标记为真的损失。损失函数定义如下:

$$loss_D = loss^{real}_D + loss^{fake}_D \quad (3-1)$$

其中, $loss^{real}_D$ 表示输入样本为真时判别模型的损失, $loss^{fake}_D$ 表示为生成样本时判别模型的损失:

$$loss^{real}_D = -\frac{1}{N_{real}} \sum_{i=1}^{N_{real}} [y^{(i)} \log D_a^{L(i)} + (1 - y^{(i)}) (1 - \log D_a^{L(i)})] \quad (3-2)$$

$$loss_D^{fake} = -\frac{1}{N_{fake}} \sum_{i=1}^{N_{fake}} [y^{(i)} \log D_{-a}^{L(i)} + (1 - y^{(i)})(1 - \log D_{-a}^{L(i)})] \quad (3-3)$$

其中, N_{real} 表示输出输入判别模型的真实样本数量, N_{fake} 表示输入判别模型的生成样本数量, $N_{real} = N_{fake}$ 。 $y^{(i)}$ 表示样本 i 输入判别模型时的期望输出:

$$y^{(i)} = \begin{cases} 1 & i \text{ 为真实样本} \\ 0 & i \text{ 为生成样本} \end{cases} \quad (3-4)$$

因此, 判别模型的损失函数可以简化为:

$$loss_D = -\frac{1}{N_{real}} \sum_{i=1}^{N_{real}} [\log D_{-a}^{L(i)}] - \frac{1}{N_{fake}} \sum_{i=1}^{N_{fake}} [\log D_{-a}^{L(i)}] \quad (3-5)$$

因为生成器和判别器进行的是二元零和博弈, 二者互为竞争关系, 因此生成模型的损失函数定义为:

$$loss_G^{real} = -\frac{1}{N_{fake}} \sum_{i=1}^{N_{fake}} [y^{(i)} \log D_{-a}^{L(i)} + (1 - y^{(i)})(1 - \log D_{-a}^{L(i)})] \quad (3-6)$$

其中, N_{fake} 为输入判别模型的生成样本数量。 $y^{(i)}$ 表示输入为生成样本时, 判别模型的期望输出, $y^{(i)} = 1$ 。因此, 生成模型的损失函数可以简化为:

$$loss_D = -\frac{1}{N_{real}} \sum_{i=1}^{N_{real}} [\log D_{-a}^{L(i)}] \quad (3-7)$$

3.4 标准模型

3.4.1 极小极大博弈

想要学习生成网络在数据 x 上的分布 P_g , 我们可以定义一个先验的输入噪声 $P_z(z)$, 然后根据 $G(z; \theta_g)$ 将其映射到数据空间中, 其中 G 为多层感知机所表征的可微函数。我们同样需要定义第二个多层感知机 $D(s; \theta_d)$, 它的输出为单个标量。 $D(x)$ 表示 x 来源于真实数据而不是 P_g 的概率。我们训练 D 以最大化正确分配真实样本和生成样本的概率, 因此我们就可以通过最小化 $\log(1-D(G(z)))$ 而同时训练 G 。也就是说判别器 D 和生成器对价值函数 $V(G,D)$ 进行了极小极大化博弈。

生成网络和判别网络可以是任意可微函数, 所以可以用梯度下降法进行迭代优化, 通常为了加快运算过程会采用随机梯度下降法 (SGD), 采用 SGD 就需要预

先建立好一个目标函数来判断以及监督学习效率。判别网络本质上是一个二分类的模型，因此可以用交叉熵代价函数来表示目标函数，即：

$$J(D) = -\frac{1}{2}E_{x \sim P_{data}(x)}[\log D(x)] - \frac{1}{2}E_{z \sim P_z(z)}[\log(1 - D(G(z)))] \quad (3-8)$$

$J(D)$ 为网络 D 的优化函数，目标函数中的 E 是数学期望， G 和 D 分别表示生成网络和判别网络的可微函数， x 是真实样本集， z 是加入的随机噪声矢量集， $G(z)$ 表示生成网络生成的伪数据集。公式 3-8 的第一项表示判别网络 D 对真实数据 x 的判别情况，第二项则表示的是对伪数据 $G(z)$ 的判别情况。 $D(x) \in [0, 1]$ ， $D(G(z)) \in [0, 1]$ ， $J(D) \in [0, \infty]$ ， $D(x)$ 越大越好， $D(G(z))$ 越小越好， $J(D)$ 越小越好。 D 网络理想最优情况下 $D(x) = 1, D(G(z)) = 0$ ， $J(D) = 0$ 。由于 G 和 D 进行的是二元零和博弈，因此生成网络的优化函数 $J(G) = -J(D)$ 。因此上述模型可以表述为如下极大极小博弈问题：

$$\min_G \max_D V(D, G) = E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))] \quad (3-9)$$

当固定生成网络 G 的时候，对于判别网络 D 的优化，可以这样理解：输入来自于真实数据， D 优化网络结构使自己输出 1，输入来自于生成数据， D 优化网络结构使自己输出 0；当固定判别网络 D 的时候， G 优化自己的网络使自己输出尽可能和真实数据一样的样本，并且使得生成的样本经过 D 的判别之后， D 输出高概率。当判别器不能区分 P_g 和 P_{data} 时，即 $D(x, \theta_d) = \frac{1}{2}$ 时，训练停止，生成网络和判别网络达到纳什均衡。

3.5 基本工作流程

生成对抗网络的训练过程并非将 $\min_G \max_D V(D, G)$ 看作优化目标，直接计算其对 θ_g 和 θ_d 的梯度，而是分为如下两个过程。

1. 训练判别模型，更新 θ_d 。将一组真实样本 x 以及一组生成样本 $G(z; \theta_g)$ 输入判别模型，计算 $\nabla_{\theta_d} \{E_{x \sim P_{data}(x)}[\log D(x)] + E_{z \sim P_z(z)}[\log(1 - D(G(z)))]\}$ ，并采用梯度上升方法更新 θ_d ，将上述过程循环 k 次。
2. 训练生成模型，更新 θ_g 。将一组一维随机变量 z 输入生成模型，计算 $\nabla_{\theta_g} \{E_{z \sim P_z(z)}[\log(1 - D(G(z)))]\}$ ，并采用梯度下降的方法更新 θ_g

训练过程中，为了让判别模型模型尽可能与生成模型匹配，即判别模型达到一定判别准确度的基础上再训练生成模型。因为真实样本并非 θ_g 的函数，因此真实样本对 θ_g 的导数为零，所以步骤 2 中计算梯度时省略了 $E_{x \sim P_{data}(x)}[\log D(x)]$ 。

3.6 本章小结

本章重点介绍了新型网络模型-生成对抗式网络。对它的生成网络和判别网络分别进行了详细的介绍和分析，然后推导出了生成对抗网络的标准博弈模型，并阐述了训练生成对抗网络的基本工作流程。

第四章 基于 GAN 图像复原的模型设计

通过以上的知识储备，将通过在朴素 GAN 的基础之上扩展并详细设计以达到我毕业论文的任务要求。

4.1 朴素 GAN 缺陷

朴素 GAN 不必根据特定的损失函数，通过就能对抗网络能够学习到比较真实的样本特征。但是，朴素 GAN 的训练不够稳定，生成的模型可能会产生没有任何含义的输出，生成的所有样本可能都在一个点上。而卷积神经网络 (CNN) 因为可以设置一系列的限制，来使模型有稳定的输出结构，是深度学习在图像处理方向运用最好的模型。鉴于以上两点，将深度卷积神经网络引入到判别模型。判别模型将输入的图像信息经过深度卷积网络后，提取图像特征，逐层减小图像的尺寸，进而输出原始图像信息的抽象表达，最后达到判别图像真伪的目的。

而引入卷积神经网络的生成模型的处理流程可以近似看作判别模型的逆过程，其目标是生成图像，将一组特征值逐层恢复成图像。生成模型将输入的一维随机变量，经过深度反卷积神经网络，通过上采样，逐层放大原始图像信息的特征，最终排列成新的图像，生成新的样本。结合了深度卷积网络和生成对抗网络的模型我们称之为 DCGAN。在此基础上引入正则化，选择对应的激活函数，取消池化，并采用新型的梯度下降函数。

1. 正则化。在判别模型和生成模型中都采用正则化，这样可以防止生成模型将所有新生成的样本都收敛到同一个点，造成生成样本多样性的匮乏；还可以将梯度传播到每一层，加速训练。但是正则化方法不能应用到判别模型的输入层和生成模型的输出层，这是因为如果正则化应用到判别模型和生成模型的所有层会影响模型的稳定性。
2. 选择对应的激活函数。生成模型除了输出层使用 Tanh 作为激活函数以外，其他的所有层次都是用 ReLU 作为激活函数；判别模型则选取 Leaky ReLU 作为所有层次的激活函数。
3. 取消池化。卷积网络池化层存在的意义是可以加速训练。在这里，生成模型使用反卷积，学习升采样；而判别模型中也可以用特定步长的卷积核，学习降采样。在使用正则化优化以及特定步长的卷积核，训练速度已经有很大提升，因此可以取消池化层。所有的池化层都改为卷积层，并且移除全连接层。

4.2 生成网络设计

生成网络如图 4.1:

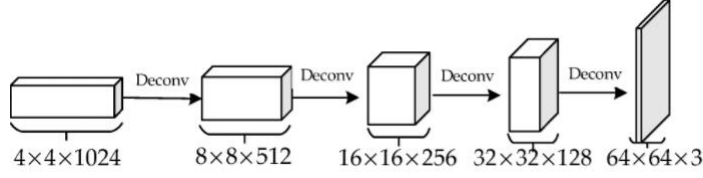


图 4.1 卷积生成网络

生成网络由一个输入层和四个隐藏层组成的五层神经网络，最后一个卷积层同时充当输出层的角色。四个隐藏层都是反卷积层。生成网络的输入是服从某一已知概率分布的多维随机序列，例如服从正态分布的随机噪声，输出是生成的样本数据。 G_L 表示神经网络的层数； $G_σ(\cdot)$ 表示第 l 层的神经元的激活函数； $G_W^l \in R^{n^{l-1} \times n^l}$ 表示第 $l-1$ 层到第 l 层神经元的权值矩阵； $G_b^l \in R^{n^l}$ 表示 l 层神经元的偏置量； $G_z^l \in R^{n^l}$ 表示 l 层神经元的输入 $G_a^l \in R^{n^l}$ 表示 l 层神经元的输出； G_norm 表示正则化函数； G_deconv 表示反卷积函数； $G_strides$ 表示卷积步长。

4.3 判别网络设计

判别网络结构如图 4.2:

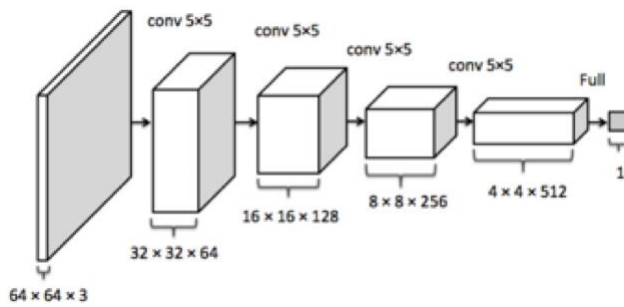


图 4.2 卷积对抗网络

判别网络和生成网络的结构基本一致，也是由一个输入层，四个隐藏层和一个输出层组成的六层神经网络结构，但是它的四个隐藏层是卷积网络。与基本的 GAN 相似，输入 DCGAN 判别模型的是真实样本或者是生成的样本，输出是当前输入是真实的样本的概率值；

D_L 表示神经网络的层数； $D_σ(\cdot)$ 表示第 l 层的神经元的激活函数； $D_W^l \in R^{n^{l-1} \times n^l}$ 表示第 $l-1$ 层到第 l 层神经元的权值矩阵； $D_b^l \in R^{n^l}$ 表示 l 层神经元的偏置量；

的偏置量； $D_{z^l} \in R^{n^l}$ 表示 l 层神经元的输入 $D_a^l \in R^{n^l}$ 表示 l 层神经元的输出； D_{norm} 表示正则化函数； D_{conv} 表示卷积函数； $D_{strides}$ 表示卷积步长。

4.4 损失函数

判别模型和生成模型都有其各自的损失函数。判别模型的末表是准确地将输入的真实样本标记为真，输入的生成样本标记为假。因此，在判别模型存在两种损失：将输入的真实样本标记为假以及将输入的生成样本标记为真的损失。损失函数定义如下：

$$loss_D = loss_D^{real} + loss_D^{fake} \quad (4-1)$$

其中， $loss_D^{real}$ 表示输入样本为真时判别模型的损失， $loss_D^{fake}$ 表示为生成样本时判别模型的损失：

$$loss_D^{real} = -\frac{1}{N_{real}} \sum_{i=1}^{N_{real}} [y^{(i)} \log D_a^{L(i)} + (1 - y^{(i)}) (1 - \log D_a^{L(i)})] \quad (4-2)$$

$$loss_D^{fake} = -\frac{1}{N_{fake}} \sum_{i=1}^{N_{fake}} [y^{(i)} \log D_a^{L(i)} + (1 - y^{(i)}) (1 - \log D_a^{L(i)})] \quad (4-3)$$

4.5 参数设置

判别网络用来完成二分类预测任务。因为卷积神经网络非常擅长图像分类，然而在改进的判别网络中缺少了池化层的采样压缩，因此只能让网络自己去学习空间下采样。该网络学习空间下采样生成网络是用来生成一个完整的图像信息。正常的卷积神经网络是在一幅图片中提取特征；这相当于生成网络的任务正好与正常的深度卷积网络过程相反，即采用的是反卷积。在图像复原过程中生成网络先进行卷积再进行反卷积复原出原始图像信息。

4.5.1 生成网络参数设置

生成的神经网络由一个输入层、四个卷积层和四个反卷积层构成， $G_L = 9$ 。不再单独具有输出层，此网络的最后一个卷积网络就作为输出层存在。

第一层，输入层。输入层的输入可以表示为：

$$G_{z^1} = [img_height, img_width, img_color_dim] \quad (4-4)$$

采用的 $img_height = 96, img_width = 96, img_color_dim = 3$ ；第二层，卷积层。卷积核：

$$G_W^2 = [5, 5, img_color_dim, 64] \quad (4-5)$$

其中卷积核的大小是 5x5，卷积核的深度是 64，初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$G_b^2 = [b_1^2 \dots b_{64}^2]^T \quad (4-6)$$

偏置量的初始化值为 0。卷积层的输入 G_z^2 为：

$$G_z^2 = G_conv(G_a^1, G_W^2, G_strides) + G_b^2 \quad (4-7)$$

其中 $G_strides = [1, 2, 2, 1]$ ，即卷积步长为 2，并且 G_conv 用补充，确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 48x48。在经过正则化和激活函数产生的卷积层输出为：

$$G_a^2 = G_{\sigma_2}(G_norm(G_z^2)) = LReLU(G_norm(G_z^2)) \quad (4-8)$$

此处激活函数选取为 leaklyReLU， G_a^2 的维度是 [48,48,64]。

第三层，卷积层。卷积核：

$$G_W^3 = [5, 5, 64, 128] \quad (4-9)$$

其中卷积核的大小是 5x5，卷积核的深度是 128，初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$G_b^3 = [b_1^3 \dots b_{128}^3]^T \quad (4-10)$$

偏置量的初始化值为 0。卷积层的输入 G_z^3 为：

$$G_z^3 = G_conv(G_a^2, G_W^3, G_strides) + G_b^3 \quad (4-11)$$

其中 $G_strides = [1, 2, 2, 1]$ ，即卷积步长为 2，并且 G_conv 用补充，确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 24x24。在经过正则化和激活函数产生的卷积层输出为：

$$G_a^3 = G_{\sigma_3}(G_norm(G_z^3)) = LReLU(G_norm(G_z^3)) \quad (4-12)$$

此处激活函数选取为 leaklyReLU， G_a^3 的维度是 [24,24,128]。

第四层，卷积层。卷积核：

$$G_W^4 = [5, 5, 128, 256] \quad (4-13)$$

其中卷积核的大小是 5x5，卷积核的深度是 256，初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$G_b^4 = [b_1^4 \dots b_{256}^4]^T \quad (4-14)$$

偏置量的初始化值为 0。卷积层的输入 G_z^4 为：

$$G_z^4 = G_{conv}(G_a^3, G_W^4, G_{strides}) + G_b^4 \quad (4-15)$$

其中 $G_{strides} = [1, 2, 2, 1]$, 即卷积步长为 2, 并且 G_{conv} 用补充, 确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 12x12。在经过正则化和激活函数产生的卷积层输出为：

$$G_a^4 = G_{\sigma_4}(G_{norm}(G_z^4)) = LReLU(G_{norm}(G_z^4)) \quad (4-16)$$

此处激活函数选取为 leaklyReLU, G_a^4 的维度是 [12,12,256]。

第五层, 卷积层。卷积核：

$$G_W^5 = [5, 5, 256, 512] \quad (4-17)$$

其中卷积核的大小是 5x5, 卷积核的深度是 512, 初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$G_b^5 = [b_1^5 \dots b_{512}^5]^T \quad (4-18)$$

偏置量的初始化值为 0。卷积层的输入 G_z^5 为：

$$G_z^5 = G_{conv}(G_a^4, G_W^5, G_{strides}) + G_b^5 \quad (4-19)$$

其中 $G_{strides} = [1, 2, 2, 1]$, 即卷积步长为 2, 并且 G_{conv} 用补充, 确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 6x6。在经过正则化和激活函数产生的卷积层输出为：

$$G_a^5 = G_{\sigma_5}(G_{norm}(G_z^5)) = LReLU(G_{norm}(G_z^5)) \quad (4-20)$$

此处激活函数选取为 leaklyReLU, G_a^5 的维度是 [6,6,512]。

第六层: 反卷积层。卷积核：

$$G_W^6 = [5, 5, 256, 512] \quad (4-21)$$

其中卷积核的大小是 5x5, 卷积核的深度是 256, 初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$G_b^6 = [b_1^6 \dots b_{256}^6]^T \quad (4-22)$$

偏置量的初始化值为 0。卷积层的输入 G_z^6 为：

$$G_z^6 = G_{deconv}(G_a^5, G_W^6, G_{strides}) + G_b^6 \quad (4-23)$$

其中 $G_strides = [1, 2, 2, 1]$, 即卷积步长为 2。此处通过上采样, 经过卷积后输出的图像的大小变为输入图像的 2 倍。此时的输出图像大小变成了 12x12。在经过正则化和激活函数产生的反卷积层输出为:

$$G_a^6 = G_{\sigma_6}(G_norm(G_z^6)) = ReLU(G_norm(G_z^6)) \quad (4-24)$$

此处激活函数选取为 ReLU, G_a^6 的维度是 [12,12,256]。

第七层: 反卷积层。卷积核:

$$G_W^7 = [5, 5, 128, 256] \quad (4-25)$$

其中卷积核的大小是 5x5, 卷积核的深度是 128, 初始参数为服从方差是 0.02 的标准正态分布。偏置量:

$$G_b^7 = [b_1^7 \dots b_{128}^7]^T \quad (4-26)$$

偏置量的初始化值为 0。卷积层的输入 G_z^7 为:

$$G_z^7 = G_deconv(G_a^6, G_W^7, G_strides) + G_b^7 \quad (4-27)$$

其中 $G_strides = [1, 2, 2, 1]$, 即卷积步长为 2。此处通过上采样, 经过卷积后输出的图像的大小变为输入图像的 2 倍。此时的输出图像大小变成了 24x24。在经过正则化和激活函数产生的反卷积层输出为:

$$G_a^7 = G_{\sigma_6}(G_norm(G_z^7)) = ReLU(G_norm(G_z^7)) \quad (4-28)$$

此处激活函数选取为 ReLU, G_a^7 的维度是 [24,24,128]。

第八层: 反卷积层。卷积核:

$$G_W^8 = [5, 5, 64, 128] \quad (4-29)$$

其中卷积核的大小是 5x5, 卷积核的深度是 64, 初始参数为服从方差是 0.02 的标准正态分布。偏置量:

$$G_b^8 = [b_1^8 \dots b_{64}^8]^T \quad (4-30)$$

偏置量的初始化值为 0。卷积层的输入 G_z^8 为:

$$G_z^8 = G_deconv(G_a^7, G_W^8, G_strides) + G_b^8 \quad (4-31)$$

其中 $G_strides = [1, 2, 2, 1]$, 即卷积步长为 2。此处通过上采样, 经过卷积后输出的图像的大小变为输入图像的 2 倍。此时的输出图像大小变成了 48x48。在经过正则化和激活函数产生的反卷积层输出为:

$$G_a^8 = G_{\sigma_6}(G_norm(G_z^8)) = ReLU(G_norm(G_z^8)) \quad (4-32)$$

此处激活函数选取为 ReLU, G_a^8 的维度是 [48,48,256]。

第九层：反卷积层。卷积核：

$$G_W^9 = [5, 5, 3, 64] \quad (4-33)$$

其中卷积核的大小是 5x5，卷积核的深度是 3，初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$G_b^9 = [b_1^9 \dots b_3^9]^T \quad (4-34)$$

偏置量的初始化值为 0。卷积层的输入 G_z^9 为：

$$G_z^9 = G_deconv(G_a^8, G_W^9, G_strides) + G_b^9 \quad (4-35)$$

其中 $G_strides = [1, 2, 2, 1]$ ，即卷积步长为 2。此处通过上采样，经过卷积后输出的图像的大小变为输入图像的 2 倍。此时的输出图像大小变成了 96x96。在经过正则化和激活函数产生的反卷积层输出为：

$$G_a^9 = G_σ_6(G_norm(G_z^9)) = ReLU(G_norm(G_z^9)) \quad (4-36)$$

此处激活函数选取为 Tanh, G_a^9 的维度是 [96,96,3]。

生成网络采用 7 层卷积神经网络，网络结构如表 4.1:

类型	卷积核	步长	输出
conv	5 x 5	2 x 2	64
conv	5 x 5	2 x 2	128
conv	5 x 5	2 x 2	256
conv	5 x 5	2 x 2	512
dconv	5 x 5	2 x 2	256
dconv	5 x 5	2 x 2	128
dconv	5 x 5	2 x 2	64
dconv	5 x 5	2 x 2	3

表 4.1 生成网络参数设置

4.5.2 判别网络参数设置

判别模型的神经网络输入由一个输入层，四个卷积层和一个输出层构成， $G_L = 6$ 。

第一层，输入层。输入层的输入可以表示为：

$$D_z^1 = [img_height, img_width, img_color_dim] \quad (4-37)$$

此处采用的 $img_height = 96, img_width = 96, img_color_dim = 3$; 第二层, 卷积层。卷积核:

$$D_W^2 = [5, 5, img_color_dim, 64] \quad (4-38)$$

其中卷积核的大小是 5×5 , 卷积核的深度是 64, 初始参数为服从方差是 0.02 的标准正态分布。偏置量:

$$D_b^2 = [b_1^2 \dots b_{64}^2]^T \quad (4-39)$$

偏置量的初始化值为 0。卷积层的输入 D_z^2 为:

$$D_z^2 = D_conv(D_a^1, D_W^2, D_strides) + D_b^2 \quad (4-40)$$

其中 $D_strides = [1, 2, 2, 1]$, 即卷积步长为 2, 并且 D_conv 用补充, 确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 48×48 。在经过正则化和激活函数产生的卷积层输出为:

$$D_a^2 = D_{\sigma_2}(D_norm(D_z^2)) = LReLU(D_norm(D_z^2)) \quad (4-41)$$

此处激活函数选取为 leaklyReLU, D_a^2 的维度是 $[48, 48, 64]$ 。

第三层, 卷积层。卷积核:

$$D_W^3 = [5, 5, 64, 128] \quad (4-42)$$

其中卷积核的大小是 5×5 , 卷积核的深度是 128, 初始参数为服从方差是 0.02 的标准正态分布。偏置量:

$$D_b^3 = [b_1^3 \dots b_{128}^3]^T \quad (4-43)$$

偏置量的初始化值为 0。卷积层的输入 D_z^3 为:

$$D_z^3 = D_conv(D_a^2, D_W^3, D_strides) + D_b^3 \quad (4-44)$$

其中 $D_strides = [1, 2, 2, 1]$, 即卷积步长为 2, 并且 D_conv 用补充, 确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 24×24 。在经过正则化和激活函数产生的卷积层输出为:

$$D_a^3 = D_{\sigma_3}(D_norm(D_z^3)) = LReLU(D_norm(D_z^3)) \quad (4-45)$$

此处激活函数选取为 leaklyReLU, D_a^3 的维度是 $[24, 24, 128]$ 。

第四层, 卷积层。卷积核:

$$D_W^4 = [5, 5, 128, 256] \quad (4-46)$$

其中卷积核的大小是 5x5，卷积核的深度是 256，初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$D_b^4 = [b_1^4 \dots b_{256}^4]^T \quad (4-47)$$

偏置量的初始化值为 0。卷积层的输入 D_z^4 为：

$$D_z^4 = D_conv(D_a^3, D_W^4, D_strides) + D_b^4 \quad (4-48)$$

其中 $D_strides = [1, 2, 2, 1]$ ，即卷积步长为 2，并且 D_conv 用补充，确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 12x12。在经过正则化和激活函数产生的卷积层输出为：

$$D_a^4 = D_σ_4(D_norm(D_z^4)) = LReLU(D_norm(D_z^4)) \quad (4-49)$$

此处激活函数选取为 leaklyReLU, D_a^3 的维度是 [12,12,256]。

第五层，卷积层。卷积核：

$$D_W^5 = [5, 5, 256, 512] \quad (4-50)$$

其中卷积核的大小是 5x5，卷积核的深度是 512，初始参数为服从方差是 0.02 的标准正态分布。偏置量：

$$D_b^5 = [b_1^5 \dots b_{512}^5]^T \quad (4-51)$$

偏置量的初始化值为 0。卷积层的输入 D_z^5 为：

$$D_z^5 = D_conv(D_a^4, D_W^5, D_strides) + D_b^5 \quad (4-52)$$

其中 $D_strides = [1, 2, 2, 1]$ ，即卷积步长为 2，并且 D_conv 用补充，确保经过卷积后输出的图像的大小变为输入图像的一半。此时的输出图像大小变成了 6x6。在经过正则化和激活函数产生的卷积层输出为：

$$D_a^5 = D_σ_5(D_norm(D_z^5)) = LReLU(D_norm(D_z^5)) \quad (4-53)$$

此处激活函数选取为 leaklyReLU, D_a^3 的维度是 [6,6,512]。接着将 D_a^5 的维度调整为 6x6x512=18432 的列向量。

第六层，输出层：输出层只有一个节点，与二分类判别器类似，输出值在 0 1 之间，用来表示输入样本属于真实样本而非生成样本的概率。第五层和输出层之间采用全连接，用 $W_{i,j}^6$ 表示第五层第 i 个神经元到输出层第 j 个神经元的权值，用 b_j^6 表示输出层第 j 个神经元的偏置量，因此 $D_z^6 = [D_W^6]^T \cdot D_a^5 + D_b^6$ ，其中 $D_W^6 = [w_1^6 \dots w_{18432}^6]$, $D_b^6 = [b_1^6]$. $D_a6 = D_z^6$ 直接作为输出。

判别网络采用 4 层卷积神经网络和 1 个全连接层，网络结构如表 4.2:

类型	卷积核	步长	输出
conv	5 x 5	2 x 2	64
conv	5 x 5	2 x 2	128
conv	5 x 5	2 x 2	256
conv	5 x 5	2 x 2	512
FC	-	-	18432

表 4.2 判别网络参数设置

4.6 算法流程

1. 加载样本集。将样本集分为训练样本和测试样本。输入训练样本是模糊的图像，输出训练样本要尽可能和对应的高清图像接近，因此样本集的每一张模糊样本都有一张与之相对应的高清样本。样本图像的规格都是 96x96 的彩色图片。
2. 训练判别模型。将真实样本的高清图像和通过模糊图像生成的样本分别输入到判别模型，采用 Adam 的优化方法进行训练，学习率设置为 0.002，重复 10 次。
3. 训练生成模型。将真实样本的模糊图像输入到生成模型，同样采用 Adam 优化方法训练模型，学习率也设置为 0.002。
4. 重复 2，3 两步，把全部真实样本迭代至平衡。

4.7 本章小结

本章重点介绍了基于 GAN 网络的图像复原方法的设计。第一部分分析了朴素生成对抗网络的训练不稳定缺陷，指出可以通过结合朴素 GAN 和 CNN 从而避免这种不稳定。第二、三、四部分，分别介绍了改进后的生成网络的设计、判别网络的设计以及损失函数的设计。第五部分对实验的网络参数设置进行了详细的说明。最后一部分则阐述了训练整个模型的算法流程。

第五章 实验及结果分析

5.1 实验环境

本次实验代码编写主要使用 python、matlab 以及 OpenMP(并行语言)。其中生成对抗网络模型的构建借助内嵌于 python 的 tensorflow 开源库。数据的预处理同样由 python 完成。实验结果的分析借助于 matlab 平台。硬件上使用了一块儿内存为 16G 的英伟达 GTX 1080Ti 显卡以及 16 核英特尔 i7 处理器。

5.2 数据准备

数据准备主要包括两个部分: 高清对照样本集和用高清对照样本集加入随机噪声后生成的模糊样本集 (即退化后的图片的集合)。我们的高清对照样本集是 Comics, 来自于斯坦福的计算机视觉实验室, 总共 50000 张彩色动漫图片, 规格统一为 96x96 的。每一张高清图片都根据规则, 生成一张于自己对应的的模糊图片。通过不同的模糊方法产生不同的模糊数据集, 并分别进行训练以及收集训练结果。其中模糊数据集包括:

- 用核大小为 [21,21] 的高斯滤波器, 对高清数据集进行 5 次迭代滤波, 产生对应的的高斯模糊数据集。
- 用尺寸为 [30,30] 的模板, 对高清数据集进行遮蔽产生对应的遮蔽模糊数据集 1
- 用尺寸为 [70,70] 的模板, 对高清数据集进行遮蔽产生对应的遮蔽模糊数据集 2

模糊样本集分割成两部分, 第一部分为训练集含有 40000 张模糊后的彩色图片, 第二部分为测试集含有 10000 张模糊后的彩色图片。

5.3 结果分析

5.3.1 高斯模糊样本集

采用核大小为 [21,21] 的高斯函数生成模糊数据集进行训练。

训练结果如图 5.1, 其中第一行图片是输入的模糊样本集, 第二行图片是第一行所对应的高清样本即目标样本集, 第三行图片是进行一次迭代训练后 G 网络复原出来的的伪图像, 第四行是进行 20 次迭代训练后 G 网络复原出的伪图像。可以发现现在进行第 20 次迭代时已经能很好地复原出原始的图像。



图 5.1 基于 DCGAN 模型训练结果

训练过程中对应的损失函数的值如图 5.2, 可以看出第 1 代的训练过程中, loss 在初期较陡, 表明模型收敛速度较快。但是第一代整体的 loss 高于 0.15, 复原效果一般。在第 10 代以及之后所有训练过程中, loss 依然存在抖动现象, 但是抖动幅度不大, 且均在 0.1 以下, 模型已经收敛到比较好的点, 结合图 7.1 可以看出复原效果已经很好。每一次迭代都会在前一次的基础上进行优化, 整体明显优于上一代, 在第 20 代基本已经达到最优。

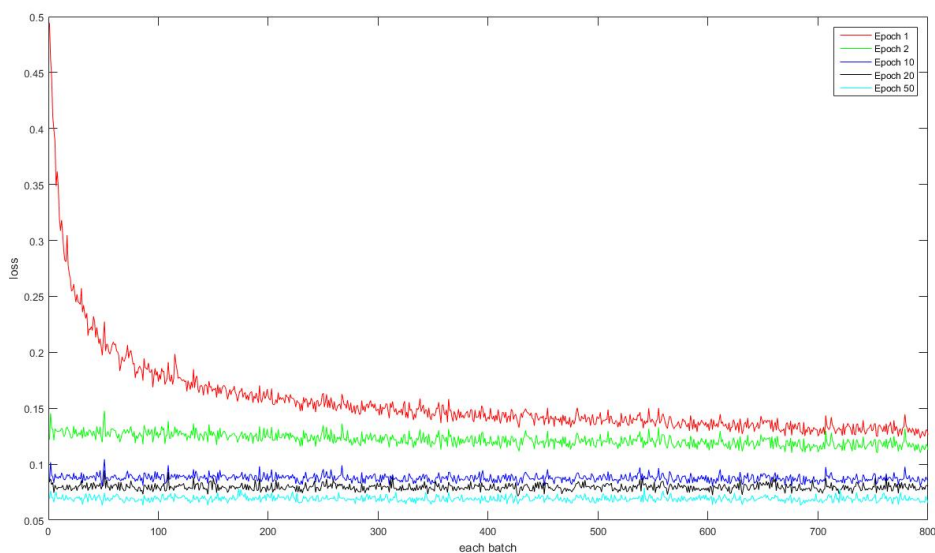


图 5.2 每一次批数据的 loss 函数值

5.3.2 遮蔽模糊样本集 1

采用核大小为 [30,30] 的模板对图像进行遮蔽生成模糊数据集。

训练结果如图 5.3, 其中第一行图片是输入的模糊样本集, 第二行图片是第一行所对应的高清样本即目标样本集, 第三行图片是进行一次迭代训练后 G 网络复原出来的的伪图像, 第四行是进行 20 次迭代训练后 G 网络复原出的伪图像。可以发现在进行第 20 次迭代时已经能很好地复原出原始的图像。且复原效果和用高斯模糊样本集训练出来的网络的复原效果差别不大。



图 5.3 基于 DCGAN 模型训练结果

训练过程中对应的损失函数的值如图 5.4, 可以看出第 1 代的训练过程中, loss 在初期较陡, 表明模型收敛速度较快。但是第一代整体的 loss 高于 0.15, 复原效果一般。在第 10 代以及之后所有训练过程中, loss 依然存在抖动现象, 但是抖动幅度不大, 均在 0.1 左右。模型已经收敛到比较好的点, 结合图 7.1 可以看出复原效果还是很不错的。每一次迭代都会在前一次的基础上进行优化, 整体明显优于上一代, 在第 20 代基本已经达到最优。

5.3.3 遮蔽模糊样本集 2

采用核大小为 [70,70] 的模板对图像进行遮蔽生成模糊数据集。

训练结果如图 5.5, 其中第一行图片是输入的模糊样本集, 第二行图片是第一行所对应的高清样本即目标样本集, 第三行图片是进行一次迭代训练后 G 网络复原出来的的伪图像, 第四行是进行 20 次迭代训练后 G 网络复原出的伪图像。可以发现在进行第 1 次迭代时并不能复原出原始的图像, 且第二幅图像的轮廓完全

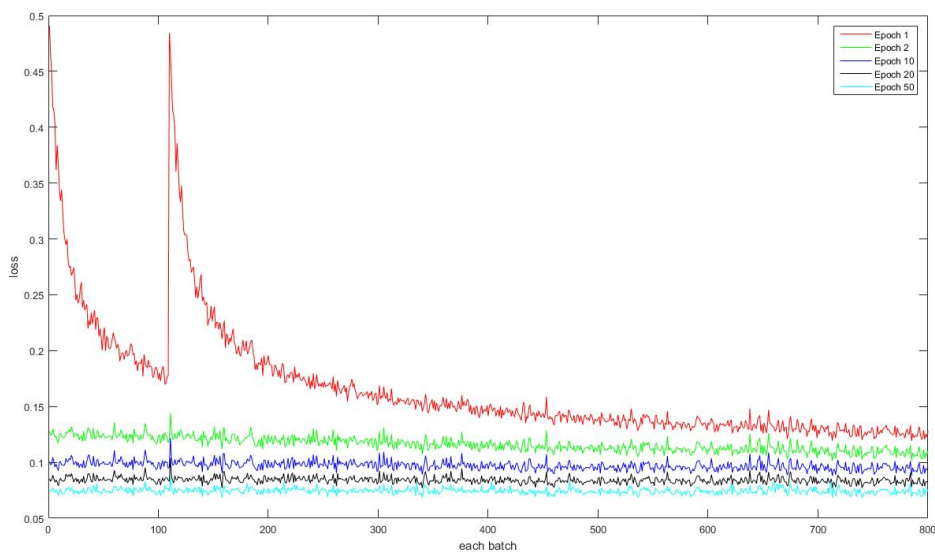


图 5.4 每一次批数据的 loss 函数值

错误。当进行到第 20 次迭代时，图像稍微有了原图的轮廓，但图像仍十分模糊。但是到第 50 次迭代，模型仍能复原出原始图像的相关轮廓，只是比较模糊。图片信息损失量超过了 53.16%，只是降低了模型的复原质量，但并不会导致复原图像出现错误，模型的鲁棒性较好。

训练结果如图 5.6, 可以看出第 1 代时，在训练过程中，loss 在初期较陡，表明模型收敛速度较快。在第 2 代以及之后所有训练过程中，loss 不会出现太大的波动，模型已经收敛。每一次迭代都会在前一次的基础上进行优化，整体明显优于上一代，在第 20 代基本已经达到完全收敛。

但是模型最终收敛的地方 loss 大于 0.1，相较于前两次的实验结果，loss 明显偏高。图片的规格是 96x96, 而遮蔽的面积为 70x70，这表明图像有效信息的一半以上都被破坏掉。因为丢失了过多的原始信息，导致本文的模型并不能达到最优。基于此当用模型进行训练时，要考虑遮蔽区域大小的问题，最好信息的损失不超过原始图像信息的 50%。当超过 50% 时，如果训练次数不是很多可能会出现复原图像信息的混淆。当训练次数达到一定规模之后，复原的图像虽然不会出现错误，也能复原出图像的轮廓信息，但质量与最优的结果也有一段距离。如第二幅图所展示的进行一次迭代，复原图片与原始图片差别较大。但进行 50 次迭代之后能复原出正确的轮廓信息但是图像轮廓比较模糊。



图 5.5 基于 DCGAN 模型训练结果

5.4 优化训练速度

1. **GPU 加速：**使用卷积生成对抗网络训练，一个最大的问题就是训练速度的问题，特别是对于深度学习而言，过多的参数会消耗很多的时间，在卷积生成对抗网络训练过程中，运算最多的是关于矩阵的运算，这个时候就正好用到了 GPU。用 CPU 的做法相比，GPU 具有数以千计的计算核心、可实现 10-100 倍应用吞吐量，因此 GPU 已经成为数据科学家处理大数据的首选处理器 GPU 本来是用来处理图形的，但是因为其处理矩阵计算的高效性就运用到了深度学习之中。tensorflow 支持 GPU 编程，但是只是对英伟达的显卡支持，而且对于 Python 编程而言，修改一些代码就可以使用 GPU 来实现加速了。
2. **并行处理：**要完成超大规模数据的训练，以及训练超大规模的神经网络，靠单 GPU 是行不通的。因为卷积对抗网络模型自身的特性，局部像素区间的独立性，数据与数据之间的独立性。因此可以考虑多个像素区间并行处理。这里的并行包括但不限于模型并行、数据并行和参数并行。

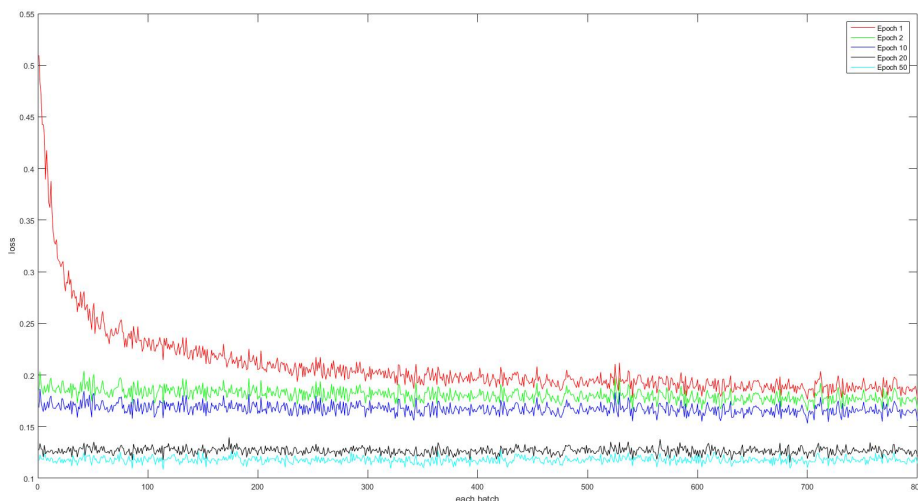


图 5.6 每一次批数据的 loss 函数值

本实验在结合 GPU 加速和多核 CPU 并行运算之后，训练时间从 50 小时降到了 4.5 小时。大大提升了训练效率，效率提高了 10 倍以上。

5.5 本章小结

本章重点阐述实验和结果分析。第一部分介绍了数据集的选取以及数据预处理的相关问题。第二部分进行了详细的实验结果分析，发现我们所设计的基于改进后的 GAN 的图像复原模型具有很好的鲁棒性和通用性。最后一部分阐述了两种优化训练速度的方法，并将两种优化方法结合到一起，使训练效率提高了 10 倍以上。

第六章 总结和展望

6.1 总结

本文主要研究了图像复原的方法，重点研究了基于 GAN 网络的图像恢复方法设计与实现。

首先，我们介绍了研究的背景、意义以及图像复原技术的研究现状。其次我们分析了图像复原的两种常用模型：基于一般退化模型的图像复原和基于卷积神经网络的图像复原，并且探讨了二者的优缺点。接下来我们研究了生成对抗网络的一般模型，为建立基于 GAN 网络进行图像复原的模型，奠定了相关的理论基础。

由于朴素生成对抗网络不够稳定，复原出的所有图像可能都在一个点上。而卷积神经网络可以通过一系列的参数限制，来使模型产生稳定的输出结构。因此本文尝试将卷积神经网络引入到生成对抗网络的生成器和判别器中，设计出卷积生成对抗网络模型，并基于此模型进行图像的复原的算法设计。本文着重介绍了在用卷积生成对抗网络模型进行图像复原时，网络结构、参数的详细设计以及算法流程。生成网络前四层为卷积层，后四层为反卷积层。判别网络只拥有四个卷积层和一个全连接层。训练过程中，为了让判别网络有更好的判别效果，我们在判别网络参数更新 10 次后，才更新一次生成网络的参数。

最后用 Comics 数据集对基于 GAN 网络的图像恢复模型进行了测试。在图片信息量损失小于百分之五十的情况下，模型能够完全地恢复图像；在图片信息量损失超过百分之五十时，模型仍然能够恢复出图像的轮廓信息。结合 loss 图像可以看出，本文设计的基于卷积生成对抗网络的图像恢复方法具有良好的收敛性、鲁棒性和普适性。

算法的处理效率和处理效果同样重要，为了提高训练的速度，在文章的最后还研究了目前最受青睐的 GPU+ 多核 CPU 的并行计算技术。并将所提出的基于 GAN 网络的图像恢复算法在并行计算平台实现。与以 CPU 串行计算为核心的系统相比，GPU+ 多核 CPU 并行运算平台可获 10 倍以上的加速比。

6.2 展望

笔者研途中主要研究了图像复原算法，总结了传统图像复原算法、卷积神经网络复原算法以及朴素生成对抗网络复原算法的优点和缺点。最后通过结合卷积神经网络和朴素生成对抗网络优点，提出了一种新的图像复原算法，并取得了较

好的结果。但研究的时间有限而所研究的数字图像处理领域却深不可测、广阔无边，自觉所研究的内容仍然有不足的地方，故而展望如下：

1. 提高可解释性: 在本文的基于生成对抗网络的图像复原模型中，在生成网络和判别网络中引入了卷积神经网络结构，直接实现了图像复原。但是这个复原的过程缺乏理论支撑，没有明显的让人可以感知到的过程特征。随着模型复杂度的提高，模型的可解释性按照同等的速度降低。模型的可解释性不论是在道德法律层面还是改进模型的层面都至关重要。近年来有人提出了 LIME 框架，LIME 作为一个更加通用的框架，旨在让机器学习的学习模型的预测更加可解释，为了保持模型的独立性，LIME 通过修改本地模型的输入来工作。因此，它不是试图同时理解整个模型，而是修改特定的输入实例，并监控对预测的影响。在图像复原的情况下，这意味着一部分像素被取代，以确定哪些像素的输入影响了图像复原的具体效果。
2. 提高通用性: 在本文的测试实验中可以发现，我们的模型对于不同样本集都可以直接进行训练，并且会产生很好的复原效果，模型的通用性已经远超一般的图像复原算法。但是我们不能忽略的是，三组实验的训练是分开进行的。每次训练，模型只用识别出单一的噪声干扰，这对于生活中更加广泛的随机噪声是有一定局限性。我们应该考虑能否把三种甚至多种噪音混合在一起，直接进行训练，尝试产生通用性更好的大模型。
3. 提高训练的速度: 本次的实验因为时间有限，只选取了 50000 张规格为 96x96 的彩色图片作为训练的样本。应用了单块内存为 16G 的英伟达 1080Ti 显卡以及多核 cpu 并行处理来加快处理的速度，进行 300 次迭代训练需要花费 4.5 个小时，这对于后期扩大数据集进行模型优化等一系列实验是一个巨大的瓶颈。因此我们可以考虑使用集群 GPU、分布式处理系统进行并行处理，进一步优化训练时间，能够期待的是这会大大缩短训练时间，提高效率。
4. 提高生成网络和判别网络收敛速度的匹配性: 在本文的实验中我们采用了更新判别模型十次后才进行一次生成模型的更新，理论上还不能判断模型的收敛性和均衡点的存在性。训练过程中要保持生成网络和判别网络的平衡和同步，否则难以取得很好的效果。然而实际过程中生成网络和判别网络的同步不容易进行，训练过程容易出现坍塌现象。如何更好的把握生成网络和判别网络收敛速度的匹配度问题，掌握生成对抗网络的收敛性和均衡点存在性的理论推断是一个值得深入研究的课题。

致谢

参考文献

- [1] 何东健, 数字图像处理, 西安电子科技大学出版社, 2003
- [2] 胡金华, 基于工件表面纹理分析的加工参数优化技术的研究, 2005
- [3] 徐瑞松, 资源环境遥感探测, 中国科学技术大学出版社, 2012 年 01 月
- [4] 左花苹, Photoshop 智能手机 APP 界面设计, 中国青年出版社, 2014 年 08 月
- [5] 孙伟, 数字图像处理系统工程导论, 西安电子科技大学出版社, 2012 年 07 月
- [6] 张高亮, 大学计算机基础, 科学出版社, 2012 年 09 月
- [7] 唐朝京, 军事信息技术基础, 科学出版社, 2013 年 01 月
- [8] 吴晓旭, 刘秀平, 刘宇, 局部加权全变差下的盲去模糊 [J]. 计算机辅助设计与图形学学报, 2014, 26(12): 2173- 2181
- [9] 薛梅, 邹采容, 杨娟, 等. 一种空间自适应正则化图像盲复原算法. 中国图象图形学报. 2002
- [10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. CoRR, abs/1409.4842, 2014
- [11] C. A. Aguilera, F. J. Aguilera, A. D. Sappa, C. Aguilera, and R. Toledo. Learning cross-spectral similarity measures with deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 1-9, 2016.
- [12] M. Brown and S. Susstrunk. Multi-spectral SIFT for scene " category recognition. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 177-184. IEEE, 2011.
- [13] A. R. YG, S. Kumar, H. Amaresh, and H. Chirag. Realtime speed estimation of vehicles from uncalibrated viewindependent traffic cameras. In TENCON 2015-2015 IEEE Region 10 Conference, pages 1-6. IEEE, 2015.
- [14] Lin Gui, Dongyin Wu, Ruifeng Xu, et al. Event-driven emotion cause extraction with corpus construction[C]. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, 1639-1649.
- [15] Zhang K, Tao D, Gao X, Li X, Li J. Coarse-to-Fine learning for single-image super-resolution. IEEE Trans. on Neural Networks and Learning Systems, 2017, 28
- [16] Du B, Wang Z, Zhang L, Zhang L, Tao D. Robust and discriminative labeling for multi-label active learning based on maximum correntropy criterion. IEEE Trans. on Image Processing, 2017, 26(4): 1694-1707.