华南师范大学

本 科 毕 业 论 文

论文题目：基于深度学习的图像信息隐藏

指导老师：苏 海

学生姓名：李 享 运

学 号：20162005048

院 系：软 件 学 院

专 业：软 件 工 程

毕业时间：2020 年 7 月

2020 年 6 月 制

基于深度学习的数字图像信息隐藏

中文摘要

随着信息技术的高速发展，人们对于信息传输的安全性、可靠性和稳定性的需求不断提高，信息隐藏技术逐渐被广泛地运用在加密传输、隐写、数字水印嵌入等方面。本文将使用深度学习中的生成对抗网络来设计隐写模型，展开数字图像信息隐藏技术（隐写术）的研究。本文将分别对生成式对抗网络的生成器和判别器进行构建。生成器经过特征提取生成嵌入概率图，经过模拟嵌入子网络生成改变图后通过结合改变图和载体图像来生成载密图像。在判别器方面，使用隐写分析模型 SRM+EC 用以判别输入图像是否有嵌入隐藏信息。通过不同嵌入率下的训练和实验结果，使用生成对抗网络来构建隐写模型可以较好地隐藏加密信息。

关键词：隐写术，深度学习，神经网络，生成对抗网络

Abstract

With the rapid development of information technology, people's requirements for the security, reliability, and stability of information transmission continue to increase. Information hiding technology is gradually widely used in encrypted transmission, steganography, and digital watermark embedding. In this paper, we will use a generative adversarial network in deep learning to design a steganographic model, and expand the research on digital image information hiding techniques (steganography). This article will construct the generator and discriminator of the generative adversarial network, respectively. The generator generates feature embedding maps through feature extraction, generates embedded change maps by simulating embedded sub-networks, and generates a covert image by combining the change map and the carrier image. In terms of the discriminator, a steganalysis model SRM + EC is used to determine whether the input image has embedded hidden information. Through training and experimental results at different embedding rates, using the adversarial network to build a steganographic model can better hide encrypted information.

Keywords: Steganography, Deep Learning, Neural Networks, Generative Adversarial Networks

目录

1. 绪论

1.1 研究背景及意义

自第三次工业革命以来，互联网和信息技术的高速迅猛发展，极大地影响了人们的生产生活方式。在不同的个体之间，信息交互的需求越来越大，对于信息传输的安全性、可靠性的要求也越来越高。由此诞生了信息加密技术，来确保通信时的信息安全。虽然数据加密可以提高数据抵抗攻击的能力，却不能隐藏通信的过程，而且加密的信息隐藏性较差，容易被攻击或拦截。因此而诞生的信息隐藏技术，可以弥补数据加密技术的不足，是当下信息安全领域十分重要的研究方向。

信息隐藏技术利用载体信息的冗余性，将秘密信息隐藏于普通信息的冗余区域之中，通过普通信息的发布而将秘密信息发布出去。它隐藏的是信息的“存在性”，使它们看起来与一般非机密资料没有区别，可以避免引起其他人注意，从而具有更大的隐蔽性和安全性，十分容易逃过拦截者的破解。信息隐藏的过程如下图所示：

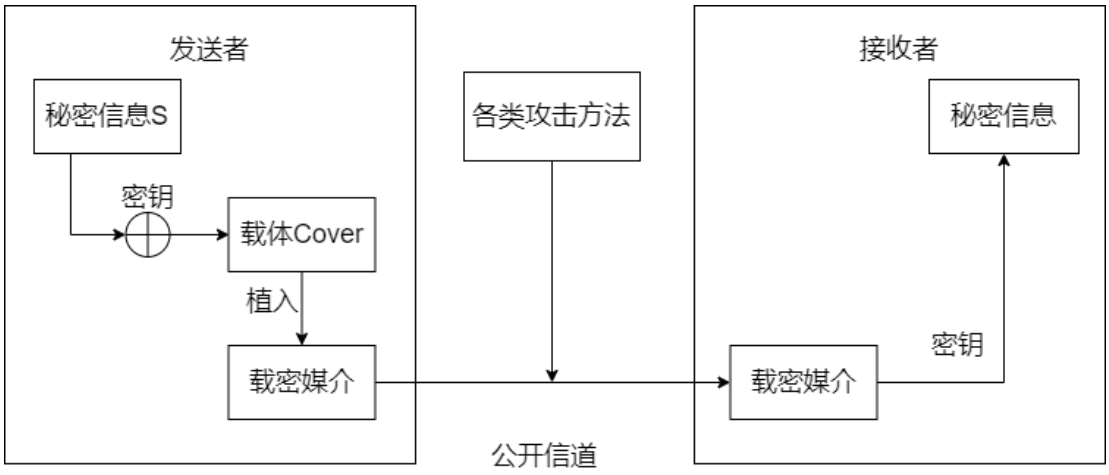

图 1-1 信息隐藏技术

发送者将不希望被他人读取的秘密信息通过密钥嵌入到载体中生成载密媒介在公开信道中进行传输。在传输的过程中，载密媒介可能会被监听者拦截，监听者希望通过各类攻击方法（包括隐藏信息分析、篡改、损毁）等方法获取或者修改秘密信息。接收者在收到载密媒介后，通过密钥重新读取到秘密信息。在这一个过程中，信息隐藏和分析算法就像是矛与盾牌的角色，相互推动对方的发展。

信息隐藏技术的衡量主要包括三个指标：隐蔽性、隐藏容量和鲁棒性。其中隐蔽性指的是秘密信息在嵌入载体后难以被发现和提取。嵌入数据的容量指的是单位在体内可以嵌入的信息多少。而鲁棒性指的是抵抗外部攻击的干扰能力。

信息隐藏技术大体可以分为：可视密码技术、数字水印技术和隐写术等。可视密码技术是将密码分为不同的持有份，然后将所有的持有份收集并按一定的顺序排列，从而得到秘密信息的技术。而数字水印技术是将信息写入信息载体媒介的技术，永久性的将信息写入载体，一旦分离载体和信息会导致载体媒介的严重损坏和无法读取，以此来保护所有者和持有者的版权。隐写术是将需要隐藏的信息写入载体媒介，通过载体进行传输从而将隐藏信息隐蔽地传递。与数字水印技术不同，隐写术的重点是要保证隐藏信息的有效性和隐蔽性，降低在通信过程中篡改、窃取等攻击的有效程度。

隐写术的隐藏信息方法，主要是将秘密信息用难以被肉眼察觉的方式隐藏到信息载体的变化较大的部分中，生成加密的载密文件，然后正常的传输给接收者。现有的隐写算法主要通过在图像的空间域或变换域进行秘密信息的嵌入但不破坏图像的基本统计特征。目前使用的较为先进的自适应型隐写算法例如S-UNIWARD[13] 、WOW [6]、HUGO[7]、HILL[8] 等算法，是通过对载体图像的纹理和噪声进行分析，找出图像变化较为复杂的区域嵌入秘密信息，以此来保证图像的特性。

但是近年来，随着互联网技术和机器学习领域的飞速发展，深度学习在许多研究的信息处理方面，得到了十分有效的应用。例如在自然语言处理、语音处理和计算机视觉等方面，取得了十分优异的效果，明显的超过了传统算法。作为隐写术的攻击算法，隐写分析算法同时也在快速地发展，优秀的隐写分析技术不断地进步，检测性能也不断地提升。目前使用的基于深度学习的隐写分析算法，可以在隐写术的参数和方法未知的前提下，通过自主学习的特征分析方法，提高隐写的分析性能。这对于信息隐藏技术带来了巨大的挑战。传统的信息隐藏方法已经开始难以对抗基于深度学习的信息隐藏分析技术。因此使用深度学习技术来革新信息隐藏技术，就显得十分的紧迫和重要。

1.2 国内外研究现状

隐写术的历史由来已久，早在古代就有通过使用特殊处理的墨水写字，书写通过特殊手法才能获取加密信息的书信，实现可见与不可见之间的转换。但是隐写术的发展过程十分缓慢，直到上世纪九十年代，基于数字媒体的隐写技术才被逐渐地开发出来，而且隐写术的目标主要是让嵌入的信息无法简单的被察觉。但是随着计算机科学和密码学的不断进步，隐写分析技术也在不断地进步，简单的隐写算法可以被成功的检测，因此促进了隐写算法不断改进和提高。总而言之，隐写技术与隐写分析技术呈现交替螺旋式的上升。

在隐写载体中，数字图像在长期以来都是十分受欢迎的载体类型，由于数字图像的存储方式不同，数字图像分为空域图像和变换域图像，因此数字图像隐写算法也分为空域图像隐写算法和变换域图像隐写算法。由于本文主要描述

和构建的是空域图像的信息隐藏模型，因此在此仅简要阐述空域的信息隐藏技术研究现状。

在空域的信息隐藏最早的方法是最低比特位替换（Least Significant Bit, LSB）[5]方法，其不但可以应用于数字图像，还可以应用在其他数字形式表示的集合内。LSB 在数字图像上嵌入的应用就是将隐藏信息使用传输的二进制符号嵌入在像素内，即用 0 或 1 代替像素值的最低有效位。LSB 算法计算速度快、嵌入信息量大，但是安全性不高。因为其安全性主要与嵌入信息的随机路径有关，如果嵌入信息的分布较为均匀，其安全性相对提高，如果嵌入信息的分布并不均匀，LSB 算法会变得较为容易被检测，甚至对隐藏信息进行篡改。如果可以人为的保证嵌入路径随机分布，LSB 匹配算法的安全性会有较大的提升。

图 1-2 LSB 方法嵌入隐藏数据对比图

但是，直接使用 LSB 嵌入，会导致图像像素颜色发生肉眼可见的变化（如图 1-2 所示），甚至是整体画面的颜色突变。在此图中，明显感觉到嵌入图的色彩相比原图要较暗。因此在将彩色图像作为隐写载体时，通常不会将隐藏信息嵌入到像素最低位，而是将信息嵌入在像素的颜色信息中。例如最优奇偶分配（Optimal Parity Assignment, OPA）算法，将颜色最为相近的两种颜色作为对应的一组，这样就可以从颜色上分配 0 和 1，达到相应的隐藏二进制信息的方法。

但是 LSB 算法对于图像的影响仍然是宏观的，对整体图像进行的直方图有显著的改变，因此 Westfeld 等[3]使用的检验方法可以很好的检测 LSB 算法。

为了提升信息隐藏的安全性，人们提出了空域下的图像自适应信息隐藏方法。自适应隐写算法最大的特点是可以根据图像内容确定隐藏信息嵌入的位置。例如将隐藏信息嵌入在纹理复杂、变化较大的地方，此时检测器就相对难以发现在上述的区域所进行的较小程度的更改。在自适应隐写算法中，通常都会用到最小嵌入失真的理念，即认为修改每一个像素都会带来不同程度的失真（可以使用标量度量）。算法在计算过程中会找到一条嵌入路径，使得嵌入信息累计嵌入的失真最小，以降低像素的修改代价。

目前的研究现状，设计优良的自适应隐写算法都可以比较准确地识别纹理复杂的区域。例如 PVD[9]、HUGO、WOW、HILL 等算法。PVD 是根据像素点对插值进行嵌入，HUGO（Highly Undetectable steGO）的失真函数设计主要是改变 SPAM（Subtractive pixel adjacency matrix, SPAM, 差分像素邻接矩阵）中影响最小的像素群来保证图像的低维统计。由于图像滤波器对识别纹理方面有着较好效果，因此自适应隐写算法也经常使用滤波器分配像素修改代价，使用高通滤波器获得纹理区域、低通滤波器获得纹理区域适当平滑并且加以扩散。HILL 利用高通滤波器和低通滤波器将秘密信息嵌入到纹理区域。WOW（Wavelet Obtained Weights）使用了多种滤波器一同计算修改代价。但是基于空域的信息隐藏算法大多数情况下比较简单，虽然嵌入信息量大，但是安全性和鲁棒性较差，特别是对图像压缩、变换等攻击抵抗力不足。

目前信息隐藏技术的研究重点主要还是在抵抗信息隐藏分析算法的攻击方面，如何根据载体图像的纹理和噪声嵌入秘密信息，成了目前研究的主要方向之一。例如 HUGO、WOW、S-UNIWARD 等算法，保证了载体图像的高阶特性的同时嵌入秘密信息，就是十分可取的做法。

1.3 主要研究内容与相关工作

本文从已有的基于深度学习的图像信息隐藏算法出发，设计了一套基于生成式对抗网络的框架，对抗训练出寻找图像适合信息嵌入的位置，生成有一定嵌入能力和抗干扰能力的空域信息隐藏方法。包括生成器的设计，判别器的设计，对抗网络模型构建等。

（1）数据预处理：从 BOSSbase1.01 数据集输入图像，经过预处理后剪裁为合适的大小。由于输入的数据集较大，经过压缩、分类等方式使其适合网络训练和应用。将数据集以 8:2 进行分割，大部分用于模型训练，小部分用于实验观察。

（2）生成器设计：使用深度学习生成输入图像的嵌入概率图，计算嵌入隐藏信息的合适位置，并模拟隐藏信息的嵌入。目标是通过深度学习计算图像纹

理变化复杂的区域用于隐藏信息的嵌入。在训练完成后，可以独立的作为生成概率图的生成器。

（3）判别器设计：判别器使用了 SRM 模型[4]，通过多种不同的空域高通滤波器，使用这些滤波器对图像滤波以获取各种类型的残差图像。然后分别统计每种相邻残差样本在一幅残差图像中出现的频次，得到残差图像的共生矩阵。最后,把共生矩阵的元素重新排成向量作为隐写分析特征。可以判别生成器输出图像和原图像是否有隐藏信息嵌入。

（4）自定义的模拟嵌入子网络：由于在生成器生成嵌入概率图后，传统图像的分类函数无法实现反向传播，而在实验过程中必须模拟隐藏信息嵌入过程，因此设计了一个模拟嵌入子网络进行模拟嵌入函数。该子网络需要一个独立的模型并进行预训练，且在整体模型的训练过程中也会不断地修正其参数。

1.4 本文结构安排

本文将介绍基于生成式对抗网络的数字图像信息隐藏技术，及其图形化界面的实现，各章内容介绍如下：

第一章：绪论。主要介绍选题背景和国内外研究现状，以及本文研究和个人工作的主要内容。

第二章：相关研究和算法综述。本章介绍了本文研究的前提知识以及机器学习的相关基础，为研究进行铺垫。包括深度学习的基本知识、基于深度学习的隐写模型和实验分析将会用到的隐写分析模型。

第三章：生成式对抗网络的搭建。通过图像预处理、生成器的构建、判别器的构建来搭建网络的基本模式和算法设计。

第四章：实验方法与实验数据分析。通过对网络的实验、测试和数据采集，与目前已有的算法进行比较、分析算法的优劣性和可用性。

第五章：总结与展望。

2. 深度学习相关研究简述

近些年来，信息隐藏领域得到了飞跃式的发展，在图像隐写和隐写分析这两个两面均有新的研究成果不断地涌现。特别是随着深度学习的兴起，深度学习因为其在学习能力、拟合能力等方面有着极强的能力受到了广泛的关注和应用。本章将先介绍深度学习的相关研究，并结合基于深度学习的隐写算法分析、比较其工作和算法的优缺点，为本文网络的提出和搭建做出铺垫。

深度学习作为机器学习的一个重要分支，其主要思想是使用包含复杂结果或者由多重非线性变化构成的多个处理层，对数据进行高层的抽象以实现最终特定的任务。深度学习的思想可以追溯到上世纪八十年代，但是由于当时的计算资源和训练数据集有限，而且在训练过程中会有十分严重的梯度消失问题，因此在当时没有得到很好的发展和引用。但是近十年来，由于计算机的算力不断加强和数据集的规模逐渐完善，深度学习的硬件条件逐渐得到满足。在另一方面，反向传播被重新重视起来，许多新模型的不断提出加快了深度学习的发展。最具代表性的有神经网络（Neural Networks），卷积神经网络（Convolutional Neural Networks, CNN）[12]、生成对抗网络（Generative Adversarial Networks, GAN）[14]、循环神经网络（Recurrent Neural Networks，RNN）[15]、自动编码器（Auto-Encoder）等。下面简要的介绍一下本文将使用到的模型结构。

2.1 神经网络

神经元是构成神经网络的最基本的元素，其源自于生物神经系统中对神经细胞的行为、特征和功能的抽象描述。

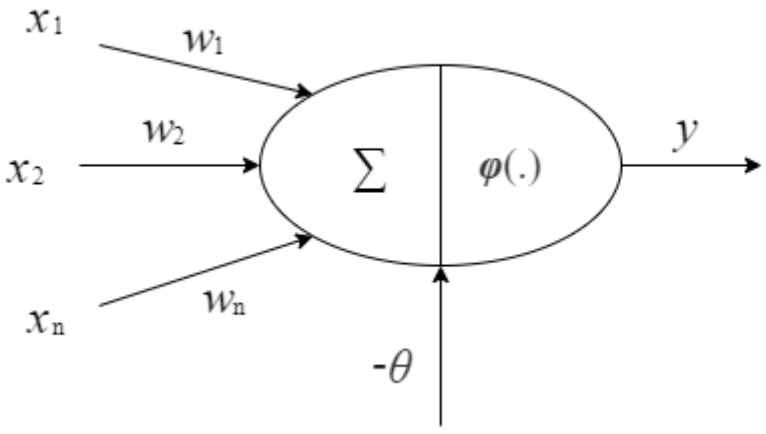

图 2-1 神经元结构

神经元结构如图所示，其中$x_1, x_2 \ldots\ldots x_n$表示神经元的输入特征，$w_1, w_2, \ldots w_n$为每个输入特征对应权值系数，$\theta$表示神经元对应阈值，表示为这个神经元被激活所需要的最小的信号量。$\varphi(.)$为激活函数，进行非线性特征表

达。从图中可以看到，神经元是一种多特征输入，单特征输出的非线性的处理单元。在这里可以对$-\theta$进行简化，将其视为多一维特征值为 1 的输入权值系数。神经元随着激活函数不同的选择，会表达出不同的输出特征，常用的激活函数有$sigmoid, tanH, ReLU$等，其公式分别如下：

$$sigmoid(s)=\frac{1}{1+e^{-z}} \tag{2.1}$$

$$\tanh(s)=\frac{1-e^{-z}}{1+e^{-z}} \tag{2.2}$$

$$\mathrm{Re}\,lu(s)=\begin{cases}s, s>0\\0, else\end{cases} \tag{2.3}$$

不同的激活函数有着不同的特征，*sigmoid* 函数在定义域内为单调递增函数，将输出映射到（0,1）的范围内，*tanh* 函数在定义域内也是单调递增函数，输出在对称的（-1,1）的范围。如果网络层数不断增加，*sigmoid* 函数与 *tanh* 函数在反向传播的时候梯度会逐渐缩小，可能出现梯度消失的情况导致网络难以收敛。于是引入 *ReLU* 激活函数，增强网络的非线性表达能力。

神经网络来源于人类对于生物神经元的数学模型构建，由一系列神经元依次、逐层相互连接构成，通过不断学习、训练、更新网络中各个神经元的参数，以此满足系统的功能。

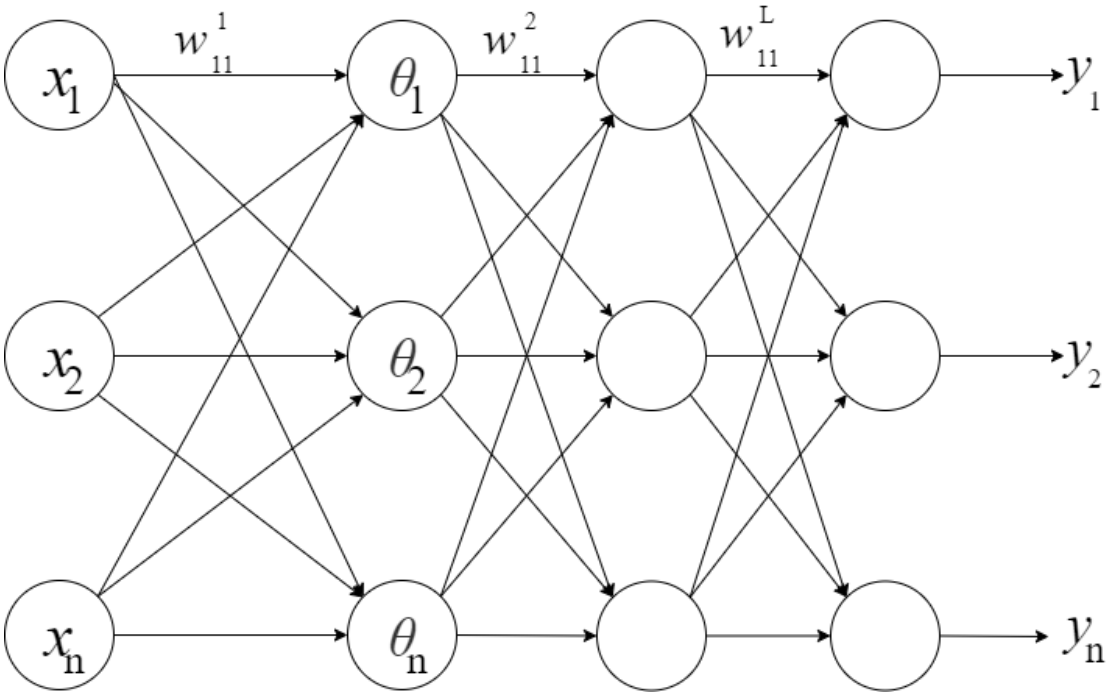

图 2-2 神经网络构造图

神经网络的结构如图所示，其中$x_1, x_2 \ldots\ldots x_n$表示网络的输入或者是上一层神经网络的特征输出，$w_1, w_2, \ldots w_n$为网络中各个神经元权值系数，$\theta$表示每个神经元的阈值，$y_1, y_2, \ldots, y_n$为网络最终输出。神经元依次排列、逐层相互连接

构成多层的神经网络，每层的神经网络输入为上一层神经网络的输出，并将本层的输出作为下一层的输入或者是最终网络输出。

2.2 卷积神经网络

卷积神经网络作为深度学习领域最具有代表性的模型结构之一，是传统感知机模型的进一步发展。卷积神经网络是一种包含多层卷积计算的神经网络。在处理图像类的问题上，通常是将图像的像素点作为参数输入到神经网络，但是如果采用是全连接的人工神经网络会导致参数过多、训练速度慢而且容易产生过拟合问题。而卷积神经网络通过局部链接、权值共享、池化操作对网络结构进行优化，如图 2-1 所示。在 CNN 的模型中，三个基本操作分别是卷积、非线性、池化。

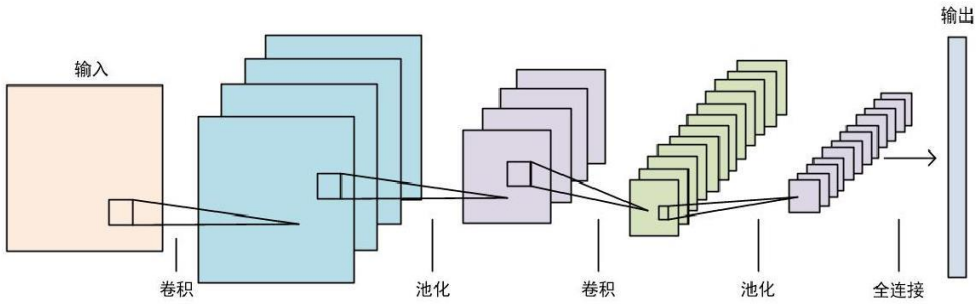

图 2-3 卷积神经网络

（1）卷积：卷积层的输入为上一层输出的特征图（feature map），用卷积核在特征图上进行相应的卷积操作，输出新的特征图。通过输入图的特征大小、卷积核的规模、卷积时的补偿和是否进行填充来共同决定输出特征图的大小；卷积核的数量决定了输出特征图的数量。这些参数在设计模型的时候由人为设定，卷积核的参数在训练之前进行随机初始化，同时在训练过程中通过反向传播不断地更新。

（2）非线性：卷积层输出的特征在通过激活函数后完成非线性操作。通常使用的激活函数包括 Sigmoid、Tanh、ReLU 等函数，这些函数极大地增强了神经网络的表达能力。

（3）池化：池化层用于统计特征图上某一位置区域的总体特征，作为 CNN 在本位置的输出。通常使用最大池化层（Max Pooling）和平均池化层（Average Pooling）。池化具有平移不变性，可以将局部区域的特征信息融合并得到更全局的特征，用于快速降低特征维度。

CNN 在计算机视觉的应用广泛关键就是在于局部链接、权值共享、池化操作以及多层次结构。局部链接是指卷积操作是卷积核仅仅与图像中一个很小区

域的像素值进行计算，所以提取到的是图像的局部特征。权值共享是一个卷积核通过滑动窗口的方式在整个图像上提取局部特征，减少模型的参数量并且提高了模型的训练速度。池化操作和多层次的结构让特征图逐渐变小，特征图的特征逐渐迭代成低层次特征组合的高阶特征。

卷积神经网络的损失传播与人工神经网络相同，都是通过计算网络输出损失按照误差反向传播算法对网络参数进行更新。

2.3 生成式对抗网络

生成式对抗网络是由 Goodfellow 在 2014 年提出的开创性的新型网络结构模型，在提出后不断有与之相关的新模型和新应用产生，使该网络的理论和模型都有着持续的发展。生成式对抗网络的基本原理是通过网络中的生成器（Generative）和判别器（Discriminative）进行对抗训练，模拟出动态博弈过程。假设网络中含有生成器 *G* 和判别器 *D*，如图 2-4 所示。

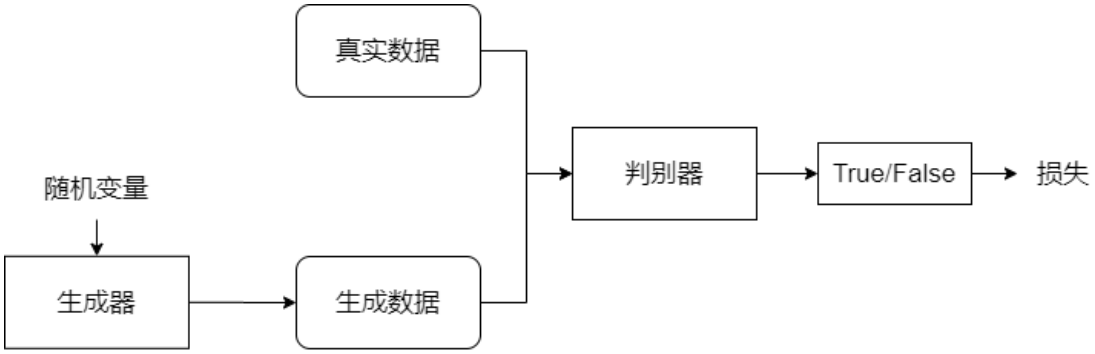

图 2-4 生成对抗网络基本模型

生成器 *G* 的任务是将输入的噪声转化为尽可能真实的数据，使得判别器 *D* 无法辨别；判别器 *D* 的任务是将生成其 *G* 输出的虚伪数据和真实数据分离。通过这个极大极小游戏，生成器 *G* 最终被训练为可以生成以假乱真的数据，判别器 *D* 无法分辨数据是否来源于生成器。

用 *E* 表示期望，*x* 表示真实数据，则有：

$$E_{x\sim p_{data}(x)}\log(D(x)) \tag{2.4}$$

当判别器 *D* 将真实数据判别为真实，即 *D(x)*=1 时，判别正确损失此时达到最大。

用 *Z* 表示为随机变量，当判别器 *D* 对生成数据判别为生成，即 *D(G(z))*=0 时，判别损失达到最大，判别器对生成数据的分类损失如下：

$$E_{Z\sim P_{data}(z)}[\log(1-D(G(z))] \tag{2.5}$$

因此，判别器的整体损失为：

$$\max E_{x\sim p_{data}(x)} \log(D(x))+E_{Z\sim P_{data}(z)}[\log(1-D(G(z))] \tag{2.6}$$

生成器的任务是训练出是能误导判别器判别的参数，因此损失与判别器相反：

$$\min E_{x\sim p_{data}(x)} \log(D(x))+E_{Z\sim P_{data}(z)}[\log(1-D(G(z))] \tag{2.7}$$

因此生成式对抗网络的最大最小化优化问题损失函数为：

$$\min_{G} \max_{D} E_{x\sim p_{data}(x)} \log(D(x))+E_{Z\sim P_{data}(z)}[\log(1-D(G(z))] \tag{2.8}$$

GAN 自从诞生以来就成为了研究热点，而且不断有新的变种出现，例如 DCGAN、WGAN[18]、LSGAN[19]等，都大大推动了 GAN 的发展并且用到了各类图像生成任务中。而生成式对抗网络中生成器 *G* 和判别器 *D* 的关系与隐写算法和隐写分析算法的关系十分相似，因此生成对抗网络用作提升隐写或隐写分析算法的模型也十分的适合。

2.4 深度卷积生成对抗网络

自从 2014 年生成式对抗网络被提出以来，虽然在相关理论和研究方面 GAN 有着巨大的发展，但是也存在着难以训练、不易收敛且模型容易崩溃，生成器输出数据质量较差等问题。为了更好的解决这些问题并且优化网络，Alec 等人将卷积神经网络和生成式对抗网络相结合，提出了深度卷积生成对抗网络（Deep Convolutional Generative Adversarial Networks, DCGAN）[16]，与之前的网络相比更容易训练并且生成的数据质量也有极大的提高。DCGAN 的生成器的网络结构图如图 2-3 所示，DCGAN 在 GAN 的基础上，优化了生成器和判别器的网络结构，在网络训练指导上进行优化。

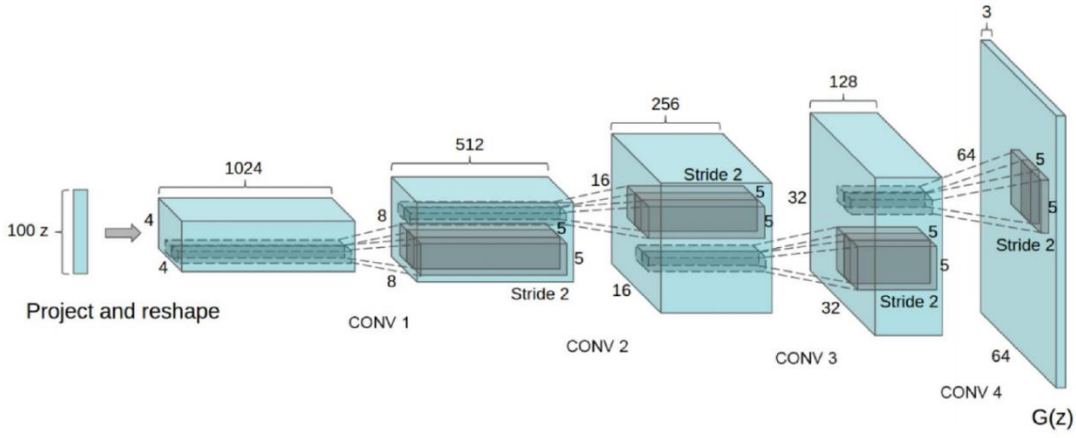

图 2-5 深度卷积生成对抗网络

在网络结构方面，DCGAN 相较于 GAN 做出了以下改进：

1. 使用步长卷积代替上采样层，卷积在提取图像特征上具有很好的作用，并且使用卷积代替全连接层。

2. 生成器 *G* 和判别器 *D* 中的几乎每一层都使用 BN 层，将特征层的输出归一化到一起。不仅加速了训练，更提高了训练的稳定性。（注：生成器的最后一层和判别器的第一层不增加 BN 层）。
3. 在判别器中使用 LeakReLU 激活函数，而不是ReLU激活函数，防止出现梯度稀疏。在生成器中仍然使用 ReLU 激活函数，但是输出层使用 tanh 激活函数。
4. 使用 adam 优化器训练，且学习率最好为 0.0002。

以上就是 DCGAN 相对于 GAN 进行的改进，可以看到 DCGAN 使用了卷积神经网络来优化 GAN 在训练过程中出现的特征提取问题。在图像处理等方面有着十分高效的应用。

2.5 本章小结

本章主要介绍了深度学习相关知识，通过介绍生成式对抗网络和卷积神经网络，阐明基于深度学习的隐写算法常用的设计思路，简要介绍了传统隐写分析算法和基于深度学习的隐写分析算法，作为本文判别器的构造思路。

3. 基于生成式对抗网络的隐写模型

本章在上章所介绍的深度学习框架基础上出发，设计了基于生成式对抗网络的隐写模型。本章在结构上分为几个部分，首先是网络构建原理，包括生成器 *G*、模拟嵌入子网络和判别器 *D*，最后是网络的训练。

3.1 网络整体构建思路

如图 3-1 所示，整体网络的构建如下：

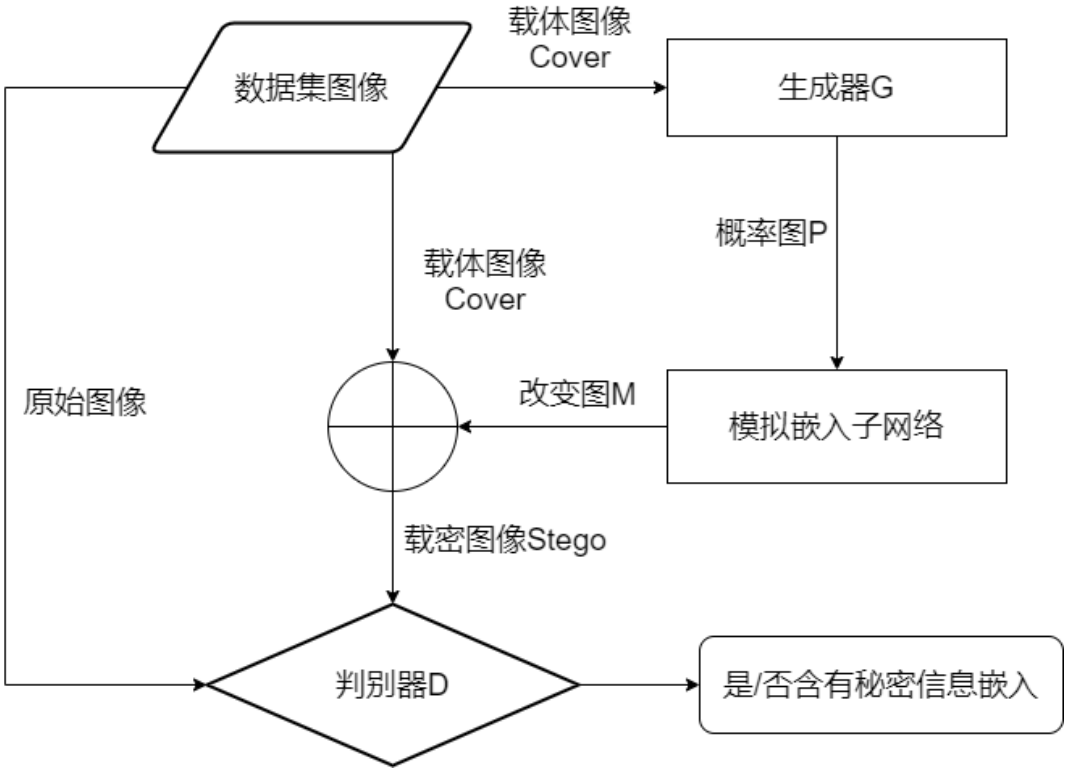

图 3-1 生成对抗网络

首先，模拟嵌入子网络作为一个独立的子网络，是需要提前进行预训练的，在下文的网络训练中会有相应的介绍。

生成器 *G* 从数据集中读取待嵌入隐藏信息的载体图像 *Cover* 经过 25 组神经网络后输出概率图 *P*。概率图 *P* 表示图像内像素适合嵌入的概率。理想的传统的自适应隐写算法，对于图像的纹理复杂、密集变化的部位嵌入概率应该较大，而平滑、变化小的部位嵌入概率应该较小，如下图所示：

图 3-2 理想的载体图像与嵌入位置的对比

生成器 *G* 的目标就是生成理想的概率图 *P*，并将其输出到模拟嵌入子网络。根据概率图 *P* 就可以算出载体图像 *Cover* 嵌入隐藏信息后需要改变的部分。之所以使用 25 组如此深层次的网络，是因为判别器 *D* 在初始状态的下的判别效果极强。因为 SRM（Spatial rich model）作为隐写分析模型本身的判别能力就十分优秀，而生成器 *G* 在起始阶段仅仅只能输出随机分布的概率图 *P*，因此判别器 *D* 对于生成器 *G* 而言判别效果过于良好，如果不使用深层次的卷积神经网络就无法达到对抗的效果。概率图 *P* 和随机生成的模拟隐藏信息结合后输入模拟嵌入子网络，得到改变图 *M*（modification map）。改变图 *M* 为隐藏信息根据概率图 *P* 输出到模拟嵌入子网络后输出的需要改变载体图像 *Cover* 的部分，通过载体图像 *Cover* 与改变图 *M* 相加后得到载密图像 *Stego*。

判别器 *D* 采用了较为成熟的基于 SRM 的模型。输入载密图像 *Stego* 和载体图像 *Cover* 混合数据集后，判别器 *D* 将判断每一张图像是来自生成器 *G* 还是来自数据集 Bossbase。通过不同的线性和非线性的空域高通滤波器对图像进行滤波操作后，可以提取出残差图像形成一定数量的子模型。每个子模型通过量化、取整和截断操作后提取出共生矩阵，再将共生矩阵合并特征矩阵后可以获取各种类型的特征向量，最后根据特征向量可以判断输入图像是否含有隐写特征。具体判别器 *D* 的实现方法在下文有详细介绍。

整体网络模型通过判别器 *D* 的判断输出，通过不断地迭代更新，优化生成器 *G*、模拟嵌入子网络和判别器 *D* 的参数，达到最终训练目的。具体的训练过程和步骤在下文中会详细介绍。

3.2 生成器 *G*

生成器 *G* 的设计思路来自于空域自适应类的隐写算法。从隐写算法的定义出发，一个成功的隐写算法应该要尽可能的缩小信息载体（*Cover*）和载密图像（*Stego*）之间的差距，用 *X*、*Y* 分别代表信息载体和载密图像，$x_{i,j}$和$y_{i,j}$分别代表其中的像素点，于是可以得到一个失真函数$D(X,Y)$：

$$D(X,Y)=\sum_{i=1}^{H}\sum_{j=1}^{W}\rho_{i,j}\,|\,x_{i,j}-y_{i,j}\,| \tag{3.1}$$

H 和 W 分别代表图像的高和宽、$\rho_{i,j}$表示像素$x_{i,j}$和$y_{i,j}$之间的改变对应的代价值。该函数可以计算出载密图像 *Cover* 和载体图像 *Stego* 在数学意义上的差距大小。因此整体的设计思路就是在学习的过程中尽可能的减少该函数的值。公式中的ρ是通过目前的自适应空域隐写算法例如 HUGO、S-UNIWARD 以及 HILL 等设计的。根据算法的计算方法不同，其嵌入代价矩阵$\rho_{i,j}$也会不同。$\rho_{i,j}$表示为该位置像素改变所需要的代价值，一般而言纹理相对简单的位置其代价值就会相对较大，纹理复杂变化较大的位置其代价值相对较小。

在给定嵌入率（单位 bits-per-pixel, bpp）和嵌入代价矩阵ρ后，由每个像素改变可能性组成的嵌入概率图$P=(p_{i,j})^{H*W}$也就可以确定了。这就是概率图 P 在传统的隐写算法中所用到的计算方法。传统隐写算法是在确定了算法后计算得到的概率 P。

从这个角度出发，利用深度学习的特点，可以将这个步骤反过来。通过深度学习确定图像的概率图P，然后再根据这个概率图得到代价矩阵ρ，其计算方法如下。

$$\rho_{\mathrm{i},j}=ln(1/\,p_{i,j}-2) \tag{3.2}$$

显然，在模拟嵌入子网络根据概率图 P 生成改变图 M 时，应当尽可能的避免在概率图$P_{i,j}$较小的位置进行嵌入操作以导致失真函数较大。关于模拟嵌入子网络的嵌入隐藏信息的流程，在下文有详细介绍。

因此生成器 G 的目标也十分的明确，就是尽可能地根据载体图像 *Cover* 生成适合的概率图 P 作为模拟嵌入子网络的输入。

生成器 G 的结构分为 25 组(group)，1~24 组的结构都相同。首先使用卷积尺寸为$7*7$的卷积核对输入的数据进行特征提取。在提取完整体特征后进行批量归一化(Batch Normalization, BN)处理，最后通过 *ReLU*（Rectified Linear Unit, ReLU）激活函数。*ReLU* 激活函数如下所示：

$$f(x)=\max(0,x) \tag{3.3}$$

在这里出现了一个问题，由于网络的层数过多，在网络中反向传播的梯度会随着连续乘而变得不稳定，有可能变得特别大或者特别小，这就是常见的梯度消失问题。使用 BN 层和 *ReLU* 函数都是为了解决相关的梯度消失的问题，但是随着网络的加深，提取的相关性实际上随着层数的增加而持续衰减。因此

这里还使用了残差网络（ResNet），将浅层网络的输出加给深层的输出，这样当网络特征达到成熟时，更深层的网络不会因为层数过多而导致输出变得不稳定。具体做法如下：

第 N（$\{N|N=2x, x\in Z\}$）层的输出特征图 F(feature map)不仅仅是输入给 N+1 层，同时，也会输出到第 N+2 层，与 N+2 层的通过 BN 层后的特征图相加。以此类推，直到 24 层为止。最后一层的输出只有一个特征图 F，其中$F_{i,j}\in(0,1)$。特征图 F 通过 *Sigmoid* 函数后将 F 矩阵所有值减去 0.5 后再经过 *ReLU* 激活函数，就可以最终输出改变矩阵 P。生成器 G 模型示意图如图 3-1 所示：

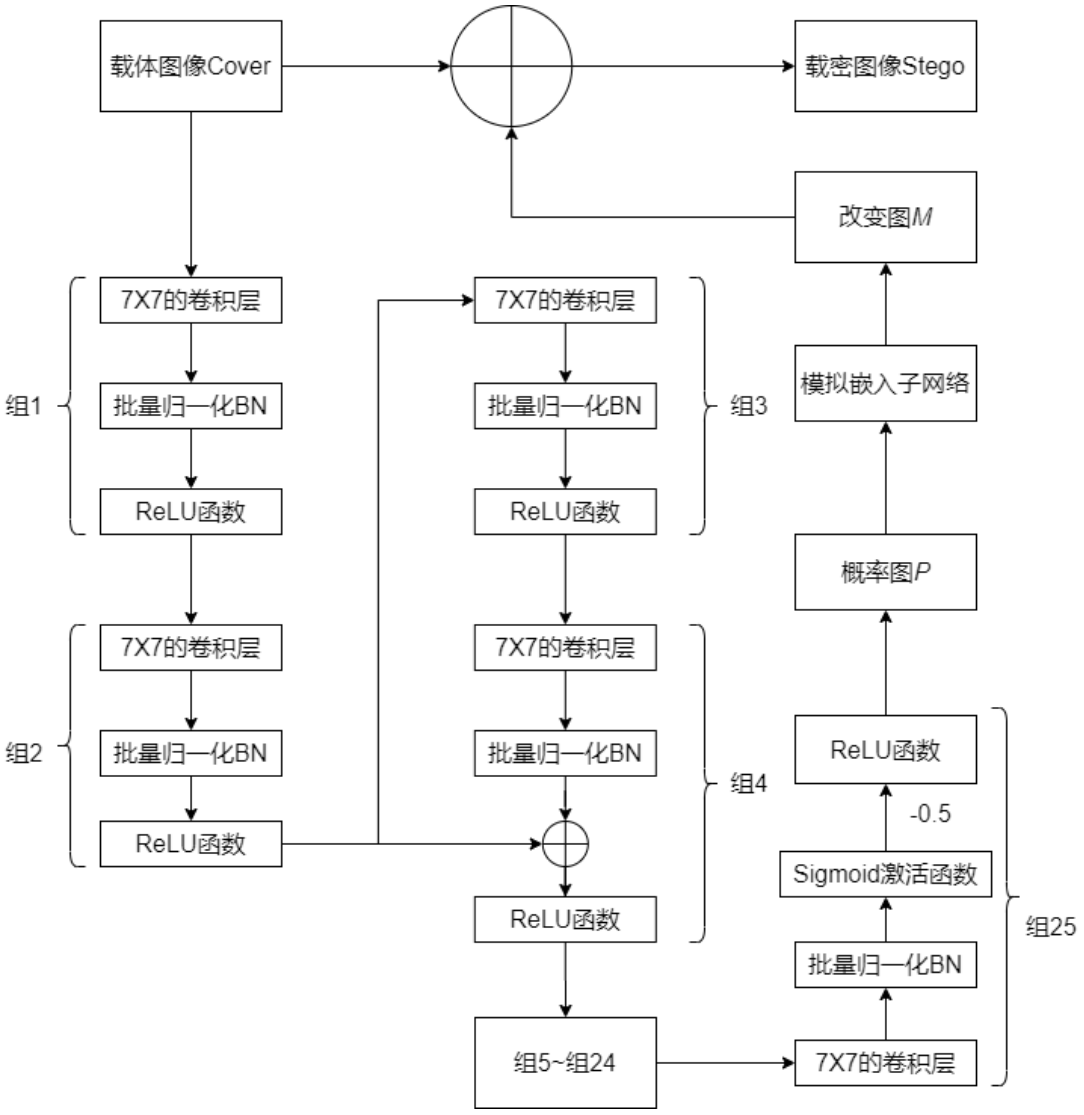

图 3-3 生成器 G 整体框架

本文将信息隐藏采用三元的嵌入模式，即对于一个像素值$x_{i,j}$其改变量有三种可能 0,1,-1，每个像素对应由生成器 G 所输出的改变矩阵概率$p_{i,j}$,因此存在如下公式：

$$p_{i,j}^{+1} = p_{i,j}^{-1} = p_{i,j} / 2 \tag{3.4}$$

$$p_{i,j}^{0} = 1 - p_{i,j} \tag{3.5}$$

所以，整体图像的嵌入率为：

$$\text{capacity}=\sum_{i=1}^{H}\sum_{j=1}^{W}(-p_{i,j}^{+1}log_2 p_{i,j}^{+1} - p_{i,j}^{-1}log_2 p_{i,j}^{-1} - p_{i,j}^{0}log_2 p_{i,j}^{0}) \tag{3.6}$$

在训练过程中，嵌入率是初始给定的，分别为 0.1bpp 或 0.4bpp。在训练过程中，每次生成概率图 *P* 时会根据概率图 *P* 计算本次的嵌入率。在理想情况下实际嵌入率应该与设定值相差不多，如果差距过大会重新生成概率图 *P*。

最后，将生成器输出的概率图 *P* 输入给模拟嵌入子网络，就得到了对应输入载体图像 *Cover* 的改变图 *M*。改变图 *M* 隐藏信息通过概率图 *P* 转化为与图像相同的大小。将改变图 *M* 与载体图像相加，就得到了载密图像 *Stego*。

3.3 模拟嵌入子网络

对于传统的自适应隐写算法，改变图 *M* 是根据改变矩阵*P*的改变概率$p_{i,j}$和一个随机数字$n_{ij} \in [0,1]$计算得到的。其中$n_{i,j}$服从均匀分布，最后得出的改变量$m_{i,j}$，具体公式下：

$$m_{i,j} = \begin{cases} -1 \ , \ \text{if} \ \ n_{i,j} < p_{i,j} / 2 \\ 1 \quad , \ \text{if} \ n_{i,j} > 1 - p_{i,j} / 2 \\ 0 \quad , \ \ \text{otherwise} \end{cases} \tag{3.7}$$

由于本文使用的是三元的嵌入方式，因此该函数同时也是一个三元的函数。在传统的自适应隐写算法中，改变图 *M* 是可以根据随机矩阵 ***N*** 和改变图 *P* 计算出来的，但是在本文所使用的深度学习框架中，直接使用（3.7）这个做法是不可行的。因为这个函数并不是连续的函数，因此存在不可导的点。虽然该函数可以由概率图 *P* 和随机矩阵 *N* 得到改变图 *M*，但是不可以进行反向传播，因此在训练过程中无法通过判别器 *D* 的输出值来改变生成器 *G* 的参数。于是需要设计一个模拟嵌入子网络来达到与公式（3.7）相同的功能，且在训练过程中可以进行反向传播。

将$p_{i,j}$和$n_{i,j}$作为输入数据对模拟嵌入子网络进行训练，训练好的模型将用以拟合上图的函数，并尽可能的给出一个相近的输出$m_{i,j}$。模拟嵌入子网络的结构如下所示：

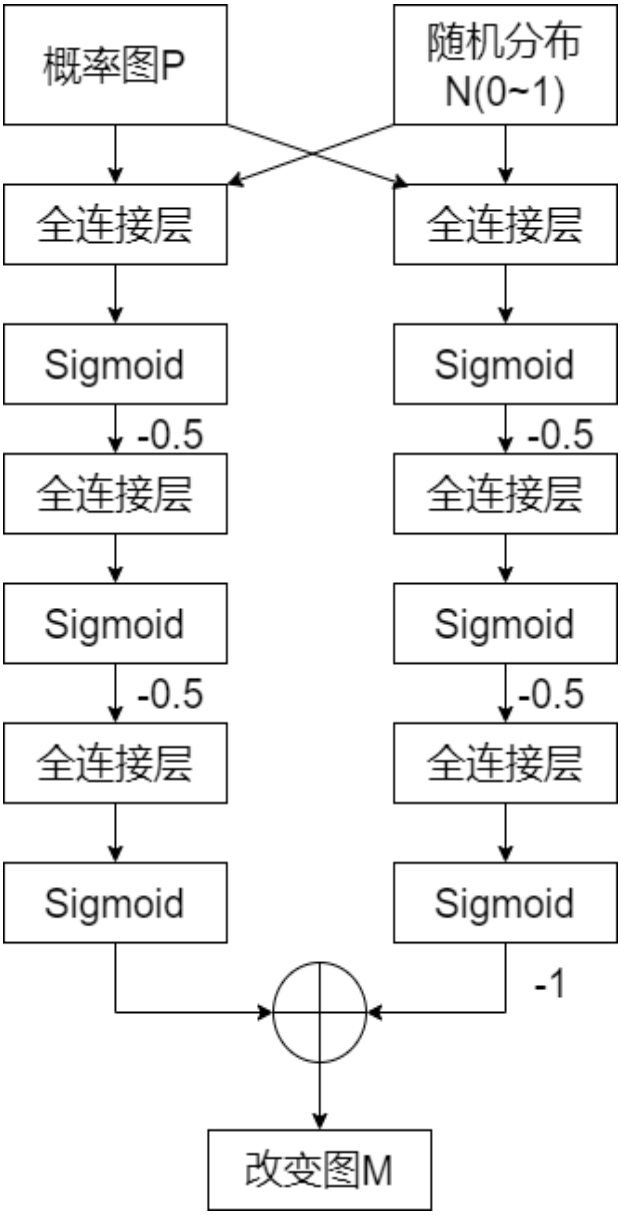

图 3-4 模拟嵌入子网络构造图

模拟嵌入子网络包含两个独立的四层全连接网络，分别称为左子网络和右子网络。每个隐藏层中包含 10 个神经元，经过 Sigmoid 函数后减去 0.5。Sigmoid 函数如下所示：

$$\sigma(z)=\frac{1}{1+e^{-z}} \tag{3.8}$$

可以看到，Sigmoid 函数的输出范围为$(0,1)$，因此在通过 Sigmoid 函数后减去 0.5 就可以使输出值由原来的以 0.5 为中心变成以 0 为中心。另外右子网络在输出后减去 1，因此数据范围变为$(-1,0)$。左子网络输出近似为 0 或 1，右子网络输出近似-1 或 0。将两个子网输出相加作为最终输出结果，于是输出值就有 1,0,-1 三种可能，这就构成了类似三元的函数，达到了与公式（3.7）类似的功能。

因此通过设计模拟嵌入子网络，不但达到了公式（3.7）的功能，可以通过嵌入概率图 P 与随机分布矩阵 N 计算出载体图像 $Cover$ 的改变图 M，同时可以满足反向传播的需求。在本文所构建的网络中核心的设计就是这一个部分。在整体网络的训练之前，模拟嵌入子网络是需要提前训练的，而且在整体网络训练的过程中还会不断完善和优化其参数。

3.4 判别器 D

判别器 D 采用空域富模型(Spatial Rich Model, SRM)[3]。该模型凭借其优越的特性提取公式被公认为传统隐写分析领域的最成功的空域隐写分析模型，配合集成学习器（Ensemble Classifier, EC）可以快速检测隐写图像。通过使用共生矩阵替代灰度直方图作为隐写特征可以有效判别某些隐写算法，比如 HUGO 算法。SRM 特征将残差与共生矩阵相结合，在准确率方面有很大的提升。SRM 特征是基于多维共生矩阵设计的，在灰度直方图上表现不明显的变化在高维特征空间中会被放大，配合使用丰富残差多方面记录微小的改动，使得特定方向的修改总是会被记录到。

SRM 特征由很多子模型特征组成，其中每个子模型计算方式大致相同，可以分为：计算残差、截短量化和共生矩阵三个部分。残差可以表示为中心像素和周围像素对中心像素预测的差值，如下所示：

$$R_{i,j} = \hat{X}_{i,j}(N_{i,j}) - cX_{i,j} \tag{3.9}$$

其中 c 表示残差的阶数，可以理解为使用邻域中像素的个数。SRM 特征一共包括六大类残差。部分残差如下所示：

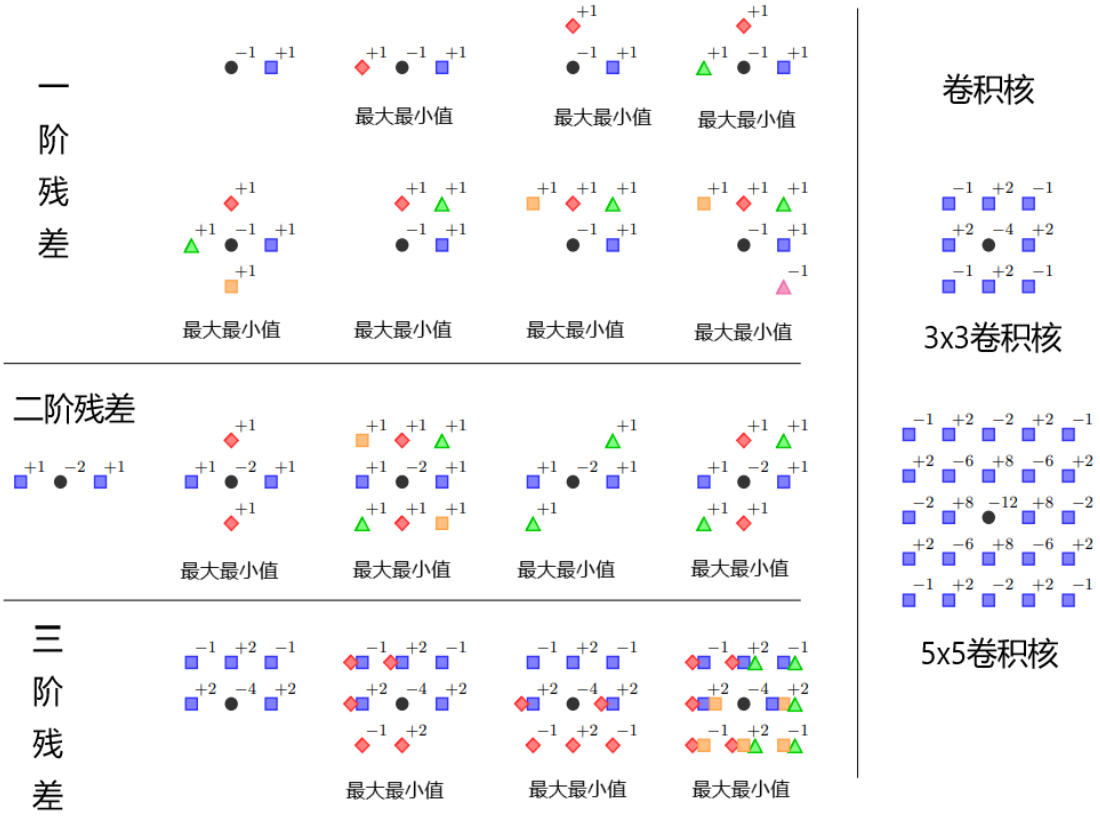

图 3-5 SRM 模型使用的部分残差

根据将图像旋转 90 度后计算的残差和为旋转图像计算的残差是否相同这个准则，将残差分为水平竖直对称残差和非对称残差。因为特征矩阵的统计是根据水平方向和竖直方向统计的，因此水平竖直对称残差统计可以得到一个共生矩阵，非对称的残差可以得到两个共生矩阵。每种残差用不同的量化步长进行量化并截短，如下所示：

$$R_{i,j} = trunc_T(round(\frac{R_{i,j}}{q})) \tag{3.10}$$

其中的 q 是量化步长，当 c=1 时，量化步长为 1 的结果和步长为 1.5 的结果几乎相同，因此只需要取两个量化步长。量化后残差按照阈值 T=2 截短，截短过程中阈值会影响共生矩阵元素的个数，阈值越大可以保留更多残差信息，但是共生矩阵中同时也会保留更多元素。不利于后续的降维。SRM 特征使用四阶的共生矩阵，按照水平方向和竖直方向进行隐性统计，相邻 4 个像素的组合个数为确定共生矩阵特定位置的值，如下所示：

$$C_d^{(h)} = \frac{1}{Z}\sum_{i,j=1}^{n1,n2-3}[r_{,j+k} = d_k, \forall k = 0,...,3] \tag{3.11}$$

其中 Z 是归一化因子，它可以使共生矩阵所有元素和为 1，当截短阈值 T=2 时，四阶共生矩阵一共含有5^4=625 个元素。通过降维操作，每一个共生矩阵可以从 625 个元素降维到 169 个元素。

以上是 SRM 的理论模型，简而言之就是通过提取不同维度的共生矩阵和残差，提取出图像不同维度的特征，以此进行隐写分析。虽然 SRM 算法根据每种残差的特征进行降维，最终可以输出高达 34671 维的特征，但是在特征的判断上，需要集成学习器选择区分效果最好的特征子集，以此作为特征分类。在实现上集成学习器训练快速且处理高位特征能力较强。

3.5 网络训练

上文就是整个模型的结构和构想，以下是训练过程。

首先是模拟嵌入子网络的预训练。模拟嵌入子网络输入的$n_{i,j}$和$p_{i,j}$都取自区间[0,1]，呈独立均匀分布。子网络中的所有参数，都由符合$(\mu = 0, \sigma = 1)$的高斯函数进行初始化，模型损失函数如下：

$$l_{TES} = \frac{1}{H}\frac{1}{W}\sum_{i=1}^{H}\sum_{j=1}^{W}(m_{i,j} - m'_{i,j})^2 \tag{3.12}$$

学习率设为 0.01，batch size 取 1000，经过10^7次迭代，模拟嵌入子网络已经能基本完成公式（3.7）的功能。另外，在整个生成式对抗网络的训练中，模拟嵌入子网络的参数还会进行修正。

在生成对抗网络的训练中，判别器 *D* 的目标是区分载体图像和生成器 *G* 所对应的载密图像，因此判别器 *D* 的损失函数如下：

$$l_D = -\sum_{i=1}^{2} y'_i \log(y_i) \tag{3.13}$$

其中y_1和y_2分别是判别器 *D* 的 *Softmax* 输出。至于生成器 *G*，其损失函数包含两部分：与生成器 *D* 对抗造成的l_G^1和确保隐写率所带来的l_G^2。

$$l_G^1 = -l_{\mathrm{D}} \tag{3.14}$$

$$l_G^2 = (capacity - H * W * q)^2 \tag{3.15}$$

生成器 *G* 的整体平均损失计算如下：

$$l_G = \alpha * l_G^1 + \beta * l_G^2 \tag{3.16}$$

在参数设定方面，$\alpha = 10^8$以及$\beta = 0.1$，这两个参数的设置都是基于l_G^1和l_G^2的大小决定的。在整体训练前，用$\mu = 0$和$\sigma = 0.01$的高斯分布初始化生成器 G 和判别器 *D* 的所有参数。

在训练数据使用方面，该模型使用了来自 Bossbase v1.01 的 8000 张图片（剩余 2000 张用于测试实验环节）进行训练。具体的数据集介绍在下文中有提到。

在训练生成对抗网络的过程中，为了提高训练模型的速度且同时提高判别器 *D* 的准确性，在每次迭代中交替地训练生成器 *G* 和判别器 *D*。同时，为了减少内存和显存的压力，在每次训练后都需要释放生成器 *G* 所生成的图像，由于每次训练时生成的概率矩阵 *P* 均不相同，因此不会出现过拟合化的现象。在训练过程中，由于生成器 *G* 的初始状态要远远弱于判别器 *D* 的状态，虽然使用了较深层次神经网络，但是在训练初期仍然不可避免出现无法生成改变图 M 的现象。因此在初始的训练中，计算嵌入率的时候并不会验算实际嵌入率，以防止整体网络结构在生成器 *G* 的部分耗费过长的时间。而在后期生成器 G 的能力逐渐提升后，将会对生成的图像的嵌入率有一定的要求，以此来加快训练速度。

训练流程如下图所示：

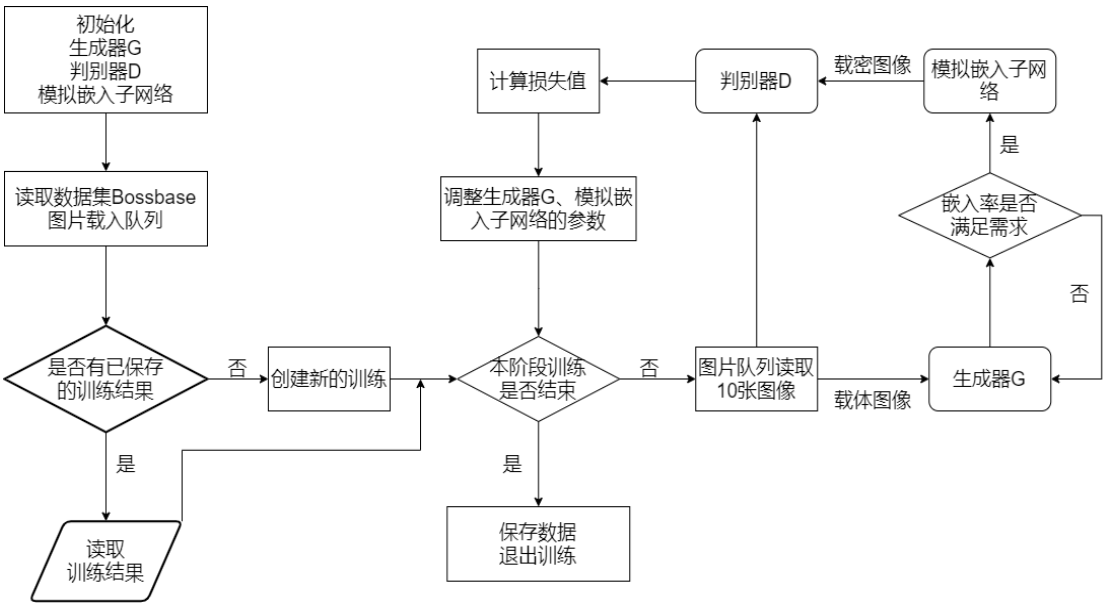

图 3-5 生成对抗网络训练流程

首先，将由生成器 *G* 生成得到 20 张图像（其中包括 10 张载密图像和 10 张原始图像）给予判别器 *D*，并使用学习率为 0.01 的随机梯度下降将l_D最小化。之后使用同样 10 张载体图像输入到生成器 *G*，并使用学习速率为1×10^{-10}的小批量的随机梯度下降法将l_G最小化。往复这个过程，以 10000 步为一个节点进行保存，在训练 30000 步后开启嵌入率检测，如果生成器 *G* 的实际嵌入率与预设值相差较大，则重新生成载密图像 *Stego*。由于本网络需要耗费大量的运算资源，因此并未设定一个实际值来决定停止训练的次数，在每 10000 步保存后会记录训练结果并做一次小规模测试。

3.6 本章小结

本章主要介绍了本文所使用的生成式对抗网络的整体构建，通过介绍生成器 *G* 的构造、模拟嵌入子网络的原理以及判别器 *D* 的构建来展示整体网络的模型。

4. 实验过程与结果分析

4.1 实验主要参数和数据

对于一个隐写算法来说，需要关注的参数主要有嵌入率，被不同的隐写分析算法的识别率和载密图像与载体图像的视觉差距。对于深度学习而言，还有比较重要的训练程度和损失值。在训练过程中，损失值会逐渐地稳定，因此之后做模型分析时，只给出最后稳定的损失值。

隐写算法的嵌入率，可以由嵌入概率图 P 计算得到，概率图 P 是由生成器 G 生成的，因此根据公式（5.1）可以计算载体图像 *Cover* 的嵌入率：

$$\mathrm{Cap}_{cover}=\sum_{i=1}^{H}\sum_{j=1}^{W}(-p_{i,j}^{+1}log_2 p_{i,j}^{+1}-p_{i,j}^{-1}log_2 p_{i,j}^{-1}-p_{i,j}^{0}log_2 p_{i,j}^{0}) \tag{3.17}$$

在实验过程中，嵌入率被固定为 0.1bpp 和 0.4bpp，分别训练出两组不同的生成器 G，在实验过程中会根据生成的概率矩阵 P 重新计算嵌入率 Capacity，记录其平均值以作为生成器 G 的训练是否有效的参数。

在载密图像 *Stego* 与载体图像 *Cover* 的比较时，使用了较为直观的对比方法，通过比较两图视觉效果上的差距以进行这一比较。总体而言，隐藏信息的嵌入是否直观，第一反应就是肉眼的感官区别。虽然也可以通过（3.1）的失真函数进行计算，但是由于对于不同的载体图像而言，嵌入代价矩阵ρ的分布和大小均不相同，而且不同的算法嵌入代价矩阵的计算方法和分布也不相同，因此无法使用失真函数作为指标。

在识别率方面，使用了基于富空间(SRM)的隐写分析模型和基于深度学习的 Xu-net 作为对比试验。使用平均被识别率作为这一项目的安全性能的指标。由于隐写分析算法有可能出现误判，因此在实验过程中取多次判定的识别率的平均值作为最后的结果。

最后是训练程度，训练程度使用训练的步数（Step）作为这一项的数据，在训练过程中会在一定的步数保存训练结果，因此该项调用起来较为便利，经过对比也可以很好地看到生成器 G 的训练程度和能力对比。

在训练和实验阶段，通过输入训练集 BossBase v1.01 优化模型和参数。本文将该数据集以 8：2 经行分割，8000 张图用于训练，2000 张图用于测试训练实验结果。该数据集来自 Agent Technology Center，在 2014 年的欧洲隐写分析算法大赛上使用的数据集，该数据集的图片来自不同型号的相机，图片均为512×512的黑白灰度图，其图片来自相机如下：

表 1 Bossbase v1.01 图片来源

图片编号	来自相机型号
1—1354	Canon EOS 400D
1355—1415	Canon EOS 40D
1416—2769	Canon EOS 7D
2770—4811	Canon EOS DIGITAL REBEL XSi
4812—6209	PENTAX K20D
6210—7242	NIKON D70
7243—10212	M9 Digital Camera

该数据集均为512×512的灰度图，内容包括各种风景、动物、静物等等，同时也包括日常环境中的抓拍。该数据集十分适合作为隐写的数据集，因为其内容大多与生活相关，许多场景在实际生活中的出现次数也较多，以下选取几个具有代表性的来源于该数据集的图片：

图 4-1 Bossbase v1.01 部分图片

该数据集通常被使用于隐写和隐写分析的相关文献和研究中，有着较好的完整性和适用性。使用 BossBase v1.01 可以很好的检测自己的隐写和隐写分析

算法对比于传统的隐写算法例如 HUGO，WOW，S-UNIWARD 有哪些不足或者区别。

4.2 实验环境及过程

本人使用的系统环境参数如下：

表 4-2 实验所用环境

硬件/软件	使用版本、容量或频率
CPU	Intel core I5-6400 4T4C 全核睿频 3.3GHz
内存	DDR4 2133MHz 16G
GPU	Nvidia GTX 1060 6G
主板型号	Notebook P7xxDM2(-G) (100 Series/C230)
硬盘	OCZ-TRION150(240GB)，THNSN5256GPU7(256GB)
操作系统	Windows 10 专业版 64 位(DirectX 12)
Python	Python 3.6.8
显卡驱动版本	441.41
Cuda 版本	V10.0
Tensorflow	1.14.0

实验通过设定不同的嵌入率：0.1bpp 和 0.4bpp 分别训练两组不同的生成器 G1 和 G2，通过与 S-UNIWARD 算法进行对比，观察其改变图 M 的嵌入分布和被 SRM 算法和 Xu-net 的检测率。SRM 算法和 Xu-net 是当前比较好的隐写分析判别器，用这两个分析器可以较好地检测隐写术的安全性能。SRM 在前文已做详细描述，不再赘述。Xu-net 是基于深度学习的隐写分析模型，在对于隐写分析方面的效果要略好于 SRM，在此作为对照组使用，观察 SRM 模型是否对生成器 *G* 的判断有所偏差。

由于不同的算法对于同一套图片所计算的失真函数不相同，因此无法使用失真函数作为比较。所以在实验过程中，同时会输入 Bossbase v1.01 中一部分图片并观察其载体图像 *Cover* 与载密图像 *Stego* 的区别作为实验对照组，以获得较为直观的训练结果。在选取对照图像的时候，不仅要观察其载体图像和载密图像在整体灰度上差距，更要观察其嵌入是否与预期相同，是否避开了纹理较为简单、清晰的区域而选择了纹理复杂变化较多的区域。

经过一定次数的迭代（如表 4-1 所示），使用生成对抗网络中的生成器 *G* 生成了 2000 张大小为512×512的嵌入了隐藏信息的载密图像。使用两个隐写分析器，包括基于 SRM[4]的隐写分析器和基于深度学习的 Xu-net[11]模型，用来评估生成器 *G* 的性能。为了排除随机数据和模型带来的可能的精度问题，对生成的数据采取多次判定后记录取平均值作为对比数据。对于生成器 *G*，同样使用了 S-UNIWARD 进行对比实验。

4.3 结果分析

首先在实验过程中收集了相关的数据，载体图像 *Cover* 和载密图像 *Stego* 的部分对比图如下：

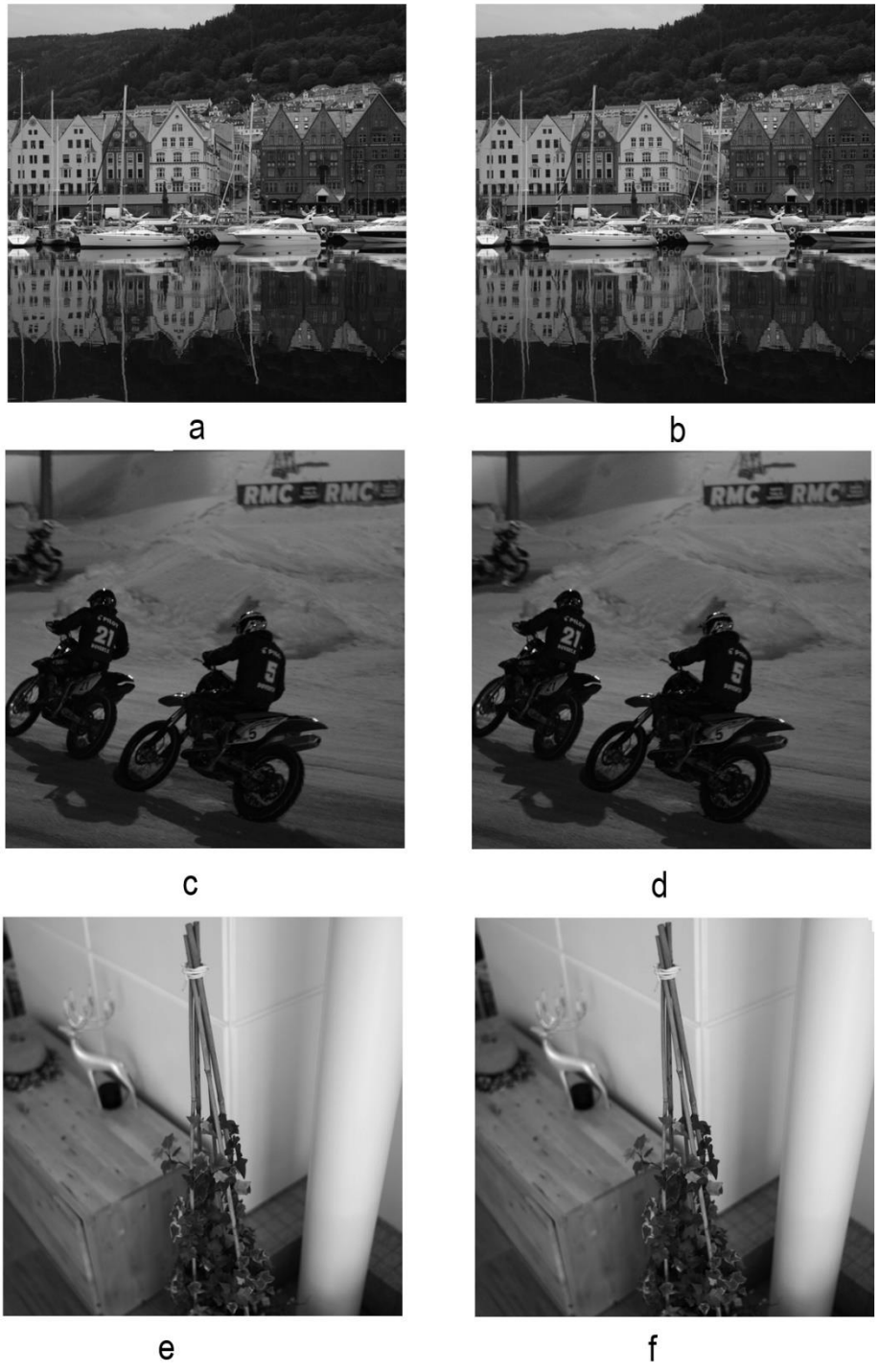

图 4-2 载体图像与载密图像对比（a）（c）（e）测试输入图，(b)(d)(f)加密图

如图 4-1 所示，左边的图均为作为载体图像 *Cover* 的原始图像，右边是嵌入了隐藏信息的载密图像 *Stego*。可以直观的对比发现，生成器的隐写效果非常的好，不论是整体的观察还是对于细节的观察，原图和载密图像都没有很大的

差别，而且在嵌入过程中也确实避开了纹理较为简单的部位（比如第三套图的右边白色部分）。没有出现整体灰度的剧烈变化和失真情况，不进行详细对比也无法看出载体图像和载密图像的区别，因此可以认为该算法生成的载密图像从直观上是比较成功的。

在训练过程中，同时采集了生成器 *G* 的嵌入率，观察其是否与实际设定值偏差过大，在设定嵌入率为 0.1bpp 时，平均实际嵌入率为 0.108bpp；目标嵌入率为 0.4bpp 时，平均实际嵌入率为 0.3972bpp。均在可以接受的误差范围内。因此可以判定生成器 *G* 的嵌入隐藏信息的方法是有效地、符合预期嵌入大小的。

同时本文在实验过程中采集了与安全性能相关的数据，如下表所示：

表 3 隐藏信息嵌入模型安全性能对比

训练步长或模型	平均未被检测率			
	0.1 bpp		0.4 bpp	
	SRM	Xu-net	SRM	Xu-net
40000	25.83%	26.49%	13.24%	9.23%
80000	27.69%	31.92%	15.49%	14.19%
120000	29.95%	36.77%	16.27%	14.69%
160000	32.52%	37.28%	16.85%	15.85%
200000	32.97%	38.19%	17.50%	16.92%
S-UNIWARD	40.15%	42.56%	21.32%	20.12%

模型的安全性能表格 4-1 所示，随着训练迭代次数的增加，生成器 *G* 的安全性能有不断地提高。为了较为直观的显示生成对抗网络在训练过程中的演化和进步，在下方给出 Bossbase v1.01 的图“08178.pgm”作为对比：

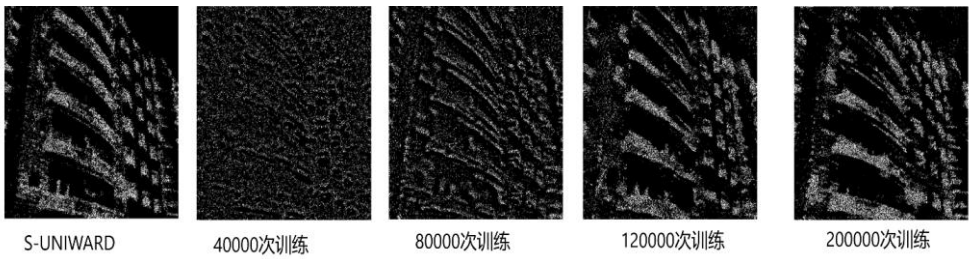

图 4-2 不同训练程度与 S-UNIWARD 的嵌入图对比

可以看到，在刚开始一直到 40000 次的迭代，生成器 *G* 不知道在什么区域更适合嵌入隐藏信息，因此整个修改图的分布几乎是随机的、均匀的分布在整个图像区域内。随着迭代训练次数的增加，生成器 *G* 逐渐学会了在图像相对较为复杂的区域嵌入隐藏信息，使得被检测出隐藏信息的概率逐渐降低。而经过了更多的训练后，生成器 *G* 的嵌入逻辑已经与传统自适应的隐写算法相当接近

了。如果使用更大更丰富的数据集进行训练，可能可以达到比传统的自适应嵌入算法更好的效果的。

如果嵌入率选择较小的 0.1bpp，可以看到该模型的安全性还是有所保证的，因为嵌入的数据量相对较少，对于隐写分析算法而言提取相应的特征也就更加的困难。如果提高嵌入率至 0.4bpp，其嵌入的安全性能也会大幅度的下降，这是由于改变的像素增多且嵌入方式较为单一的原因。因此在实际使用的时候应该相应的减少嵌入的信息量。

但是不得不承认，即使经过大量的训练，与 S-UNIWARD 算法相比，在准确性和安全性上都有着一定差距，目前对于这种情况有着以下几种猜测：

（1）： 实验所使用的样本集不足。由于 Bossbase v1.01 仅有 10000 张左右的灰度图，而且还仅仅使用了其中 8000 张作为训练，所以在数据输入方面相对较为单一。如果使用更大更为复杂的是数据集，可能会取得更好的结果。

（2）：实验环境相对较弱：对于本文所设计的网络而言，本人所使用的实验环境硬件性能较弱，导致卷积层运算效率极低，不得不降低 Batch size 以保证模型能正常训练。

总而言之，从实验的结果来看，本文所设计的网络在数字图像隐写方面是较为成功的，可以完成隐藏信息嵌入到载体图像且不为肉眼所察觉。在实验过程中的实际嵌入率和预设值的差距也在可允许的误差范围内。在未被检测率方面虽然略低于 S-UNIWARD 算法，但是可以看到随着训练的不断深入和规模逐渐扩大，本文模型的隐写方法的安全性有着不断地提高。

4.4 本章小结

本章从实验中所关注的参数入手，简明的阐述了参数列表，实验所需环境和实验过程，最后介绍了实验的结果分析，包括生成对抗网络的隐写效果和与传统隐写算法的对比。可以看到生成对抗网络确实可以用来实现隐写模型，效果虽然不及传统算法，但是使用深度学习来进行隐写是确实可行的，在经过大量的训练后可以得到近似于传统算法的水平。

5. 总结与展望

随着互联网技术和深度学习的不断发展，监听和窃听的情况不断增多，解密和分析算法也逐渐完善，对信息隐藏技术来说是十分巨大的挑战。本文对基于深度学习的信息隐藏算法展开研究，首先从时代背景和传统算法入手，介绍了传统算法的不足和对本文研究的展望，然后从深度学习的基础知识、生成式对抗网络的构建、隐写算法的本质展开研究，最后实现了基于生成式对抗网络的隐写模型，惊醒了实验和数据分析。

在生成器 *G* 方面，本文从隐写的定义和分析出发，通过生成概率图逆向推测出改变图 *M*。在生成器 *G* 的编码中，使用了 25 层卷积神经网络，取得了一定的效果。在判别器 *D* 方面，使用了简要介绍的 SRM 模型，取得了较好的判别效果，对生成器 *G* 的训练起到了良好的推动作用。在解决嵌入函数无法反向传播的问题上，使用了模拟嵌入子网络，通过进行预训练取得了较好的效果。在整体网络构建网面，通过生成器 *G* 生成概率图 *P*，使用模拟嵌入子网络计算出改变图 *M* 最后与载体图像 *Cover* 相加生成载密图像 *Stego*，使用判别器 *D* 判断混合了载密图像 *Stego* 和载体图像 *Cover* 来训练生成器 *G*。最后通过数据集选择 Bossbase v1.01，具有较好的稳定性和客观性，也具有足够大的数据规模，方便训练和对比实验。

经过实验和数据分析表明，本文基于生成式对抗网络的隐写模型可以进行隐写操作，通过对载体图像 *Cover* 嵌入隐藏信息的方式生成载密图像 *Stego* 进行隐藏信息的嵌入。在被检测概率方面具有一定的抵抗力、抗攻击能力，整体的综合效率也较高。

但是本文同样还存在很多不足之处。首先是该生成器基于深度卷积神经网络构建，神经网络的层数过多，训练效率十分的低下。如果没有较好的设备，需要很长时间的训练，损失值才会趋于稳定，不是很适合实际的使用场景。其次是数据集的选择较为单一，仅仅使用了 8000 张512×512的灰度图作为训练，如果使用更多更为复杂的训练集，应该可以得到更好的效果，但是由于上文说到的训练效率低的问题，选择更多更大的数据集会造成训练效率更加的低下，因此本文的网络和算法还有许多需要改进的地方。最后是本文提出的模型，在测试过程中展现出的数据表明，其并没有比传统的算法有更好的效果，仅仅是证明使用深度学习的技术来进行隐写是可行的，在被识别率方面弱于 S-UNIWARD 算法，证明了这个模型确实还有需要优化的方面。

未来基于本文类似的研究，应该可以从生成器 *G* 的模型入手，设计更合理的网络模型来使得概率图的生成更加合理。

参考文献

[1]. FRIDRICH J,弗里德里希,张涛,等.数字媒体中的隐写术:原理,算法和应用[M]. 北京:国防工业出版社, 2014.

[2]. Lecun Y , Bottou L , Bengio Y , et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[3]. Westfeld A , Pfitzmann A . Attacks on Steganographic Systems[J]. 1999.

[4]. Fridrich J , Kodovsky J . Rich Models for Steganalysis of Digital Images[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(3):868---882.

[5]. Patel K , Utareja S , Gupta H . Information Hiding using Least Significant Bit Steganography and Blowfish Algorithm[J]. International Journal of Computer Applications, 2013, 63(13):24-28.

[6]. Filler T , Fridrich J . Gibbs Construction in Steganography[J]. IEEE Transactions on Information Forensics & Security, 2010, 5(4):705-720.

[7]. Vojtěch Holub, Fridrich J . Designing Steganographic Distortion Using Directional Filters[C]// IEEE Workshop on Information Forensic and Security. IEEE, 2012.

[8]. LI B , WANG M , HUANG J , et al. A new cost function for spatial image steganography[C],IEEE International Conference on Image Processing, 2014: 4206–4210.

[9]. Wu D C , Tsai W H . A steganographic method for images by pixel-value differencing[J]. Pattern Recognition Letters, 2003, 24(9-10):1613-1626.

[10]. Tang W , Tan S , Li B , et al. Automatic Steganographic Distortion Learning Using a Generative Adversarial Network[J]. IEEE Signal Processing Letters, 2017, PP(99):1-1.

[11]. Xu G , Wu H Z , Shi Y Q . Structural Design of Convolutional Neural Networks for Steganalysis[J]. IEEE Signal Processing Letters, 2016:1-1.

[12]. Lecun Y , Bottou L , Bengio Y , et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[13]. Vojtěch Holub, Fridrich J , Tomáš Denemark. Universal Distortion Function for Steganography in an Arbitrary Domain[J]. EURASIP Journal on Information Security, 2014, 1(1):1.

[14]. I. Goodfellow et al., “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst. [J] 2014, pp. 2672–2680.

[15]. Siegelmann H T . Recurrent neural networks[M]// Computer Science Today. Scholarpedia, 2006.

[16]. Radford A , Metz L , Chintala S . Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J]. Computer Science, 2015.

[17]. Qian Y , Dong J , Wang W , et al. Deep learning for steganalysis via convolutional neural networks[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2015, 9409:94090J-94090J-10.

[18]. Arjovsky M , Chintala S , Bottou, Léon. Wasserstein GAN[J]. 2017.

[19]. Mao X , Li Q , Xie H , et al. Least Squares Generative Adversarial Networks[J]. 2016.
[20]. Odena A . Semi-Supervised Learning with Generative Adversarial Networks[J]. 2016.
[21]. Shi H , Dong J , Wang W , et al. SSGAN: Secure Steganography Based on Generative Adversarial Networks[J]. 2017.
[22]. Filler T , Judas J , Fridrich J . Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes[J]. IEEE Transactions on Information Forensics and Security, 2011, 6(3):920-935.
[23]. Yang J, Liu K, Kang X, et al. Spatial Image Steganography Based on Generative Adversarial Network[J]. 2018.
[24]. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation[J]. 2015.
[25]. Jamie Hayes, George Danezis. Generating Steganographic Images via Adversarial Training[J]. 2017
[26]. Jiren Zhu,Russell Kaplan,Justin Johnson. HiDDeN: Hiding Data With Deep Networks[J]. //The European Conference on Computer Visvion(EVVC), 2018
[27]. Bas P , Tomáš Filler, Tomáš Pevný. ”Break Our Steganographic System”: The Ins and Outs of Organizing BOSS[C]// Proceedings of the 13th international conference on Information hiding. Springer Berlin Heidelberg, 2011.
[28]. 董士琪. 基于生成对抗网络的图像信息隐藏研究[D]. 2019.
[29]. 岳普. 基于深度学习的数字图像隐写算法研究[D]. 2019.
[30]. 任科. 基于深度学习的图像信息隐藏方法研究[D].2019.

致谢

本论文是在我的指导老师苏海老师的悉心指导下顺利完成的。在这里首先要感谢苏海老师对我的关心和指导。从论文的选题到生成对抗网络的实现，离不开老师对我的帮助和鼓励。苏海老师不论是对待课堂还是科研，都展现了十足的敬业精神和一丝不苟的态度。在大学期间参加的各类比赛，也离不开苏海老师的大力支持。同时还要感谢在大学期间对我帮助和影响颇深的潘家辉教授，在大学的科研项目方面和竞赛方面给予了我极大地帮助，潘老师对待学生的态度也令我十分的感动。

同时，我还要感谢我的母亲，是她在这么多年的生活中对我悉心的照料和关心，在我低迷沮丧的时候给予我关心和动力，在我获奖高兴时与我一同欢笑。她为了家中的老老小小辛苦工作，让我有能力完成学业。

最后，要感谢我的同学、朋友在我大学期间给予的鼓励和帮助。特别是在比赛场上与我一同奋斗的陈广源同学和黄泳锐师兄，赛场上的一同努力，平时的练习都离不开相互的支持，感谢他们在算法竞赛上给我带来的动力。同时也要感谢软件协会的同学们，从无到有的软件协会，是离不开各位的努力和帮助的，在协会大家一同学习一同进步的时光真的非常令人怀念。

大学本科四年学习就要接近尾声，我即将告别校园进入社会。非常感谢在华南师范大学软件学院度过的四年时光，不仅仅学会了软件工程的相关专业技能和只是，还教会了我做人做事的道理。在这里祝软件学院的未来一片光明。