华 中 科 技 大 学

本科生课程考试答题本

考生姓名 邬雪菲

考生学号 U202112131

专　　业 网络空间安全

考试科目 多媒体数据安全

考试日期 2024 年 5 月 24 日

评　　分

题型	得分	题号	得分
论文报告			
实验报告			

总分：	评卷人：

注：1、无评卷人签名试卷无效。

2、必须用钢笔或圆珠笔阅卷，使用红色。用铅笔阅卷无效。

目 录

一、研究背景

随着人工智能技术的飞速发展，语音识别系统在安全关键型应用中扮演着越来越重要的角色。自动语音识别（ASV）技术，作为验证特定话语是否由特定人发出的手段，已经被广泛应用于从金融服务到智能家居安全等多个领域。近年来，深度学习技术的引入极大地推动了 ASV 的进步，涌现出了大量高性能模型，能够以极高的准确度识别和验证说话者的身份。

然而，ASV 系统同样有着机器学习模型的普遍弱点，即容易受到对抗性攻击的威胁。对抗性攻击是一种通过在输入样本中引入精心设计的微小扰动，来误导机器学习模型的技术。这些对抗样本在人类感知上与真实样本相似，但对高性能模型却能够产生欺骗效果。在语音识别领域，这意味着攻击者可以通过对抗性音频样本来操纵 ASV 系统，比如通过误导自动语音识别（ASR）进行恶意命令的注入，或通过欺骗说话者识别（SR）将攻击者错误地识别为已注册的用户。现有的对抗性音频攻击研究主要集中在向原始音频添加小扰动来制作对抗样本。这些扰动通常是在 Lp 范数的约束下进行优化的。但人为引入的噪声往往会留下不自然的、可识别的人工痕迹，这不仅易被人类察觉，而且可以被用来防御对抗性攻击。

为了解决这一局限性，本论文提出了一种新的对抗音频攻击方法——语义相关对抗音频攻击（Semantically Meaningful Adversarial Audio AttaCK，SMACK）。与传统的基于噪声的对抗样本不同（如图 1），SMACK 专注于修改语音的固有属性，例如韵律，这是一种代表语义属性的特征。通过这种方式，SMACK 能够在不降低语音质量的前提下，生成能够在语义上代表相同语音的对抗性音频样本。这样不仅能够逃避现有的基于噪声检测的防御机制，而且由于其高度的自然性和隐蔽性，对 ASV 系统的安全性构成了新的挑战。

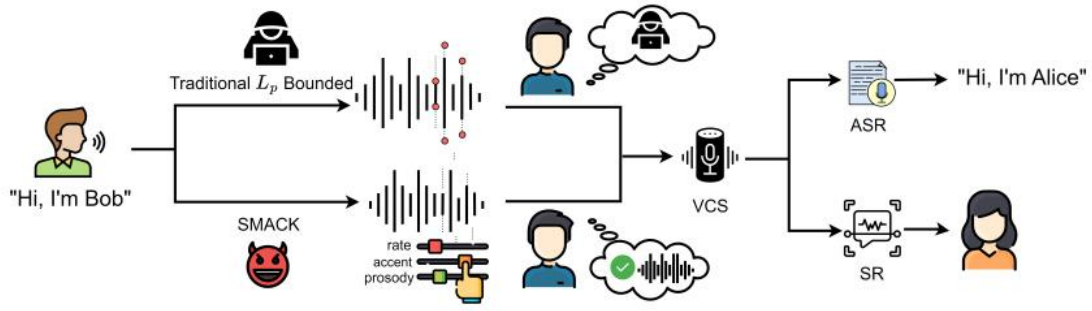

图 1 SMACK 与传统对抗音频攻击比较

二、方法介绍

SMACK 的核心思想是通过修改语音的语义属性来生成对抗性音频样本，这些样本对人类而言听起来自然，但能够逃避最先进的防御系统。SMACK 利用了一个适应性的生成模型来控制韵律，该模型接受一个连续空间中的向量来实现韵律的控制。为了优化复杂的韵律特征，本文开发了一种结合了扩展的遗传算法和梯度估计的新颖算法。同时，针对不同威胁对象，SMACK 有着不尽相同的攻击方式。

下面将从语义属性建模、威胁模型、攻击方式三个板块具体介绍。

2.1 语义属性建模

语义属性的概念首先在图像领域中引入，作为一种数据增强方法来减轻过度拟合。关键是沿着语义方向转换线性化特征空间中的数据样本，从而产生与具有相同类别标识但不同语义的另一个样本相对应的特征表示。

保留语义的视觉修改是有效的，但对语义相关的音频扰动却知之甚少。图像中的语义属性由像素表示，这些像素需要与层次对象关联和颜色描述相关的空间相关性，而音频波形的语义属性例如韵律则采用时间序列的格式，具有时间依赖性。图像中的空间语义扰动可以用来攻击识别模型，但时间语义修改是否可以产生类似的效果仍然未知。

2.1.1 语义属性选择

虽然音频领域的一些现有工作提出了用于数据增强的语义属性的概念，包括韵律、重音和词序，但将它们用于对抗样本生成的可行性尚未被探索。本文研究指明，对于用于隐秘对抗攻击的语义特征，它必须满足三个关键要求：

（i）固有属性：在音频领域，固有属性是指不应该依赖于语音内容或说话者身份而是应该广泛存在于几乎所有语音中的属性。典型的固有属性包括情感、语速、口音、韵律等。

（ii）身份内容保留：这个要求在于从人类的角度保留原始标签。对于 ASR 系统，语义对抗性音频样本的内容应该与人类的原始内容相同。同义词替换单词无法满足此要求。对于 SR 系统，语义对抗性音频样本和原始音频的身份信息应该相同。

（iii）自然性：除了保留原始身份之外，还强调语义扰动的自然性。直观

上，虽然语义属性是人类语音的固有属性，但较大程度的修改会产生人工痕迹，使人类来听起来不正常。

本文利用韵律作为代表性的语义属性。韵律通常被描述为语调和节奏。它包含语音的多个特征，如语音的音高轮廓或语调、音节的长度、单词的响度等，是由多种频域和时域描述符描述的复杂属性。

选择韵律是因为它满足语义对抗样本生成的三个要求。首先，韵律是人类语音最重要的固有特征之一，已在多个领域得到广泛研究，包括情感识别、语音合成和语言研究。其次，与口音和词序等其他语言特定特征不同，韵律不仅限于语音内容，而且具有普遍适用性。第三，韵律可以通过每个帧中的细粒度特征来表示，并且在语义上限制韵律的操纵可以保留内容和自然性。

2.1.2 韵律建模与控制

由于韵律的复杂性，其控制仍然是一个开放的研究问题。为了解决这个问题，本文设计了一个适应的生成模型，用于逐帧细粒度的韵律控制。下图 2 显示了生成模型设计的架构，包含四个主要组件：内容网络、韵律网络、韵律内容交叉注意力模块和解码器网络。

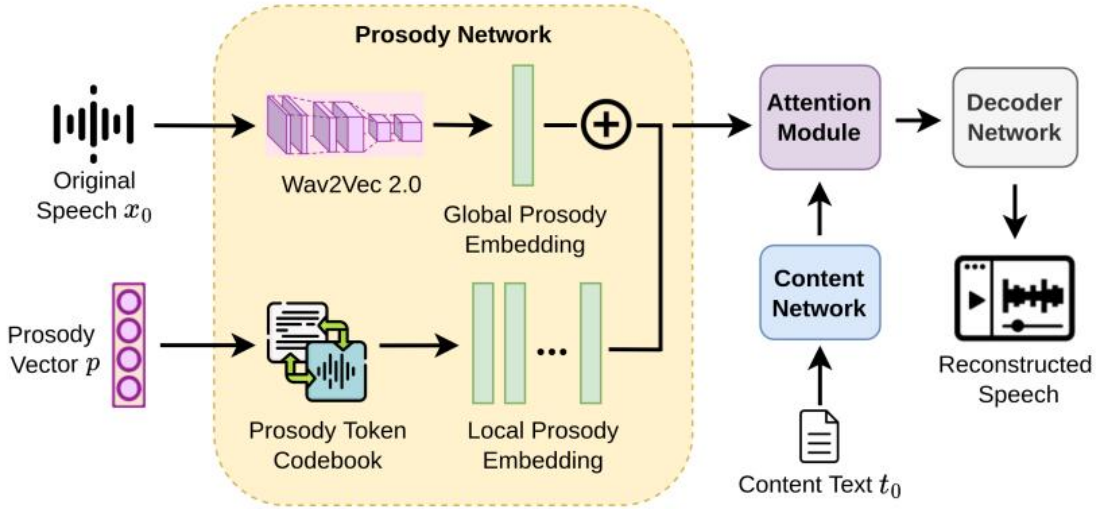

图 2 韵律生成模型结构

（i）内容网络：使用原始语音的文本来约束重构语音的内容（即语义对抗性音频样本）与原始语音相同。

（ii）韵律网络：生成的音频的韵律由全局韵律和局部韵律两个组件定义。

全局韵律代表了每个说话者独特的语气，因此表征了说话者的身份。它由通过特征提取器从原始音频中提取的全局韵律嵌入来表示。为了满足身份保留要求，

SMACK 中的全局韵律嵌入保持不变。

局部韵律侧重于帧级韵律特征，它是由可变长度输入韵律向量（即攻击者旨在优化的向量）和可训练韵律标记码本的矩阵乘法产生的。输入韵律向量具有可变长度，因为其丰富度（即，韵律向量 p 的维度）是可配置的并且取决于语音内容（例如，语音内容的长度）。韵律表示是通过在时间维度上广播添加全局和局部韵律嵌入来获得的。

（iii）注意力模块和解码器：将来自上下文网络和韵律网络的信息对齐以获得融合嵌入，该嵌入将用于解码器网络重建语音音频。在这种情况下，生成语义相关的对抗性音频样本相当于优化控制对抗性操作的韵律向量 p。

2.1.3 韵律优化机制

韵律向量 p 的可变长度和黑盒设置给优化带来了独特的挑战。为了应对这些挑战，本文提出了一种两阶段优化机制。它由自适应遗传算法（AGA，Adapted Genetic Algorithm）和梯度估计方案（ES，Estimation Scheme）组成，即 AGA-ES 算法。第一阶段（AGA）执行全局搜索，旨在通过优化韵律向量的长度和值来减少搜索空间。第二阶段（ES）作为细粒度的局部优化方案，针对对抗目标和语音自然度细化韵律。值得注意的是，本文所提出的 AGA-ES 框架可以适用于攻击具有不同损失函数的 ASR 和 SR 系统。

（i）自适应遗传算法

遗传算法是一种基于遗传和自然选择原理的基于搜索的优化技术。它通过应用包括选择、突变和交叉在内的遗传算子来工作，已被证明在多个应用领域与深度强化学习具有竞争力。

在 SMACK 中，两个独特的优势使其非常适合本文的优化问题。首先，韵律向量的可变长度和无界值导致很大的搜索空间，遗传算法已被证明在大范围搜索中是有效的。其次，攻击是在黑盒设置下开发的，遗传算法在这种情况下效果很好，因为它不需要梯度信息。

在 SMACK 中，p 的潜在解被建模为染色体，因此每个基因代表韵律向量中的一个值。然而，标准遗传算法不能直接应用于 SMACK，因为搜索空间是刚性的，这意味着遗传算子中染色体的长度是固定的。基于变长遗传算法的概念，引入了一种名为插入和删除（InsDel）的新算子，该算子在生物染色体中普遍存在，如

下图 3 所示：

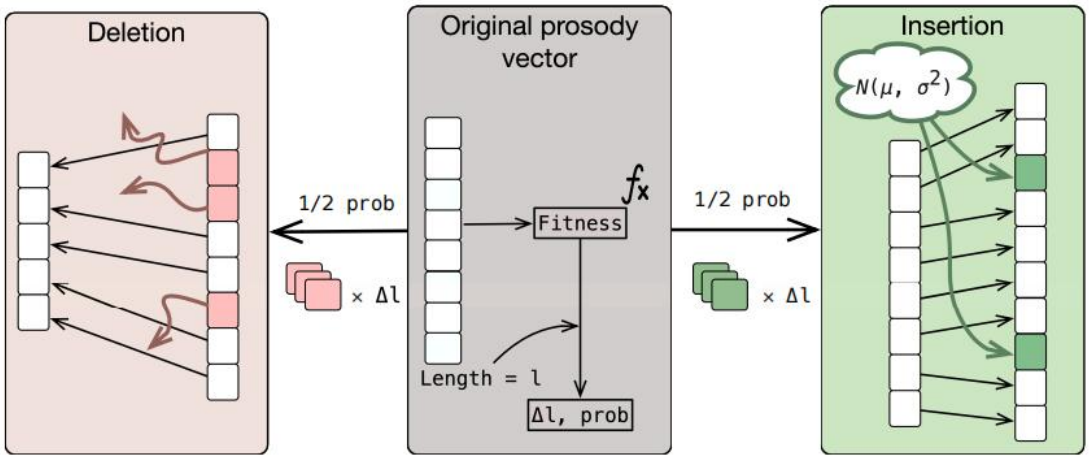

图 3 InsDel 算子的基本概念

一部分基因会按照自适应概率被添加到原始染色体上或从原始染色体上去除。操作概率、插入与删除、操作位置、编辑长度、插入值这五个因素用来表征操作的有效性。首先，将 InsDel 的概率设计为与适应度值和分数提高呈负相关，从而实现收敛速度和适应度提高之间的平衡。其次，插入和删除被设计为以相同的概率发生，以引入无偏的长度变化。第三，随机选择插入/删除的基因以引入额外的值变化。第四，编辑长度被设计为取决于迭代次数和原始韵律控制向量，以提高自然度和运行时间。最后，为了自然性，插入的值是从原始韵律向量的分布中采样的。

（ii）梯度估计方案

尽管适应的遗传算法在大空间探索中效果很好，但在接近最优解时效果较差。因此，通过调整梯度方向引导的韵律来补充优化。在此阶段不改变韵律向量的大小，仅细化其值。当考虑攻击者无法访问目标模型梯度的黑盒设置时，通过称为自然进化策略（NES）的梯度估计方案来近似第 i 次迭代的梯度。它的工作原理是沿着各个方向向韵律向量添加噪声，并将梯度近似为由损失值加权的平均方向。

2.2 威胁模型

SMACK 旨在操纵原始音频的韵律，导致 ASR 和 SR 系统的错误分类。针对 SMACK 对语音安全系统产生的威胁，有如下假设和建模：

攻击目标：攻击者旨在对 ASR 或 SR 系统进行有针对性的攻击。在针对 ASR 的攻击中，对抗性音频应被转录为与人类解释不同的单词/句子。对于针对 SR

系统的攻击，攻击者的目的是制作和播放对抗性音频，以便 SR 算法将其误识别为来自已注册的说话人。

攻击者假设：假设攻击者仅拥有黑盒知识（既无法访问目标模型的架构，也无法访问目标模型的参数），并且对音频源的访问有限。对于 ASR 攻击，假设攻击者只能访问转录，因此它被认为是硬标签黑盒攻击。对于 SR 攻击，假设攻击者有权访问最终结果（接受/拒绝）和置信度得分，还假设攻击者无权访问系统中注册用户的语音样本，因此他无法使用 DeepFake 创建听起来像受害者的音频样本。

目标系统假设：假设 ASR 和 SR 系统被配置为提供最佳的识别率，并且识别模型随着时间的推移保持不变，还假设商业 API 和 VCS 为同一平台提供的服务是相似的。

对抗样本生成和交付：对抗性音频样本提前生成，并且可以通过 API 在线传送到目标或通过播放传输到 ASR/SR 设备。假设攻击者可以播放整个对抗性音频样本而不是其中的一部分。

2.3 攻击方法

2.3.1 攻击语音识别系统 ASR

将目标 ASR 系统表示为 ASR(x)，它将音频 x 作为输入并返回其转录内容。攻击者的目标是生成一个语义对抗样本 x*，使得：

$$ASR(x^*)=t,\text{s.t. } x^*=G(t_0,x_0,p),t\neq t_0,H(x^*)=H(x_0)$$

其中 t 是攻击目标，H(•)表示将音频 x 解释为他/她感知的文本的人。

为了实现攻击目标，攻击者的目标是通过最小化由两个分量组成的多目标损失函数来获得对抗样本：

$$\mathcal{L}=\text{argmin}_{x^*}\{L_{ASR}(ASR(x^*),t)+wQ(x^*)\},$$

其中 L_{ASR} 是将生成的音频推向对抗转录的对抗项，引入 Q(•)来衡量重建音频的人类可解释性，其中 w 作为权重因子，在对抗目标和攻击隐蔽性之间进行权衡。

下图 4 给出了基于 AGA-ES 的语义对抗样本生成的完整算法。首先构建了一个适应 SMACK 的多目标损失函数 L，其中提出了一种新的距离函数，结合了编辑

距离和发音相似度的测量，并引入了 NISQA 评分作为保留自然性的质量监控因素。

Algorithm 2: SMACK against black-box ASR

Input: Prosody control vector p
Original speech x_0 with content t_0
Target transcription t
Maximum number of iterations N

Output: Adversarial audio example x_{adv}

pop $\leftarrow [p] *$ popSize
function $Fitness$(pop, t)
    popAudio $\leftarrow G(t_0, x_0,$ pop$)$
    fitness $\leftarrow L_{ASR}(ASR($popAudio$), t) + \alpha Q($popAudio$)$
    return fitness
for i in range(N) **do**
    fitness $\leftarrow Fitness($pop$, t)$
    bestPop $\leftarrow$ pop$[Argmax($fitness$)]$
    if $Fitness($bestPop$, t) <$ Threshold **then**
        topPop $\leftarrow Select($pop, fitness$)$
        for j in range(childSize) **do**
            newPop $\leftarrow InsDel($topPop$)$
            newPop $\leftarrow Mutate($newPop$)$
            newPop $\leftarrow Crossover($newPop$)$
        end for
    else
        bestFitness $\leftarrow Fitness($bestPop$, t)$
        probePop $\leftarrow$ [bestPop] $* K + \sigma \boldsymbol{u}$
        gradEst $\leftarrow \frac{1}{\sigma K} \sum Fitness($probePop$, t) \times \mathbf{u}$
        bestPop $\leftarrow$ bestPop $+ \eta \cdot sign($gradEst$)$
    end if
end for
get x_{adv} generated by $G(t_0, x_0,$ bestPop$)$
return x_{adv}

图 4 针对黑盒 ASR 的 SMACK 算法

具有编辑距离的对抗项：对抗项旨在测量当前转录与目标短语之间的距离，这使得算法能够相应地调整优化策略。该术语通常源自标签上的损失值或概率分布。然而，如威胁模型中假设，此类信息在实践中通常无法获得。因此，对抗性术语只能使用最终转录的硬标签来构建。编辑距离已被用于计算通过一系列单字符编辑（例如插入、删除或替换）将一个字符串转换为另一个字符串的最小成本

来测量两个句子之间的差异。然而，这样的测量在攻击中是不足的，因为它只考虑了字符编辑。例如，“book”→“back”的编辑距离为 2，与“book”→“ok”的编辑距离相同。然而，“back”和“book”在发音上具有较高的相似度，这使得通过操纵语音属性（即韵律）将 ASR 混淆为错误分类相对容易。造成这种差异的根本原因在于编辑距离独立对待单词中的每个字符，而忽略了发音因素，而发音因素通常取决于字符组合和相关性。

具有语音相似性的对抗性术语：由于使用编辑距离作为攻击中唯一度量的局限性，因此转向音系学和语言学，试图根据构成音素来量化距离，其中音素是一个单位能够区分单词的声音。音素可以通过字素到音素转录器直接从转录中获得。尽管如此，仅仅对音素应用编辑距离有其局限性，原因有两个。首先，元音和辅音对语音的贡献不同。因此，用元音替换辅音应该测量更大的距离，而不是用另一个元音替换它。其次，一些音素彼此表现出更多的相似性，因此这些音素之间的任何替换都会产生更少的成本。以 ARPAbet 为例，音素“EM”→“EN”替换操作的测量距离应该比“EM”→“V”的测量距离更小，即使这些音素都是辅音。

基于这些见解提出了一种改进的编辑距离，其中每个音素对的编辑成本都分配有自定义的权重。遵循现有语音研究中的假设，即当编辑操作频繁发生在同一单词的两个替代发音之间时，编辑操作的成本较低。因此，收集从 CMU 发音词典中提取的单词的不同发音的音素，并采用 Needleman-Wunsch 算法以最小编辑距离对齐每对发音。然后，通过出现频率的统计分析来实现音素相似度 S(a,b)，S(a,b)值越大表示音素 a 和 b 之间的相似度越高。

然而，这种相似性是从替代发音发展而来的，专注于提取相似的音素，因此对于具有显着差异的音素对（例如元音和辅音）来说是不完整的。通过基于 ALINE 同源对齐系统计算音素差异来补充这一点，其中音素是用一组根据其显着性加权的特征 F 来描述的特征对相似性的影响。差异测量如下：

$$D(a,b)=\sum_{f\in\mathbf{F}} df(a,b,f)\times salience_f+|V(a)-V(b)|$$

$$V(phoneme)=\begin{cases} v_{cst}, & \text{if } phoneme \text{ is a consonant,} \\ v_{vwl}, & \text{otherwise.} \end{cases}$$

其中 df(a,b,f)计算给定特征 f 的 a 和 b 之间的差异，v_{cst} 和 v_{vwl} 是确定辅音和元音相对权重的启发值。最后需要结合前面提到的编辑距离，以避免由发音非常相似但转录不同的单词引起的局部极小值。

编辑距离和语音相似度结合,对抗性术语可以表述为：

$$L_{ASR} = w_1 \frac{Leven(t^*, t)}{Len(t^*) + Len(t)} + w_2 D(t_p^*, t_p) - w_3 S(t_p^*, t_p)$$

其中 $t*_p$ 是转录文本 t*的音素构成,Len(·)返回字符串输入的长度,w1,w2,w3 是因子权重。

在损失函数中引入了质量评估术语来评估转换后的语音的质量和自然度。传统的 Lp 范数和信噪比 (SNR) 不足以解决问题，因为它们仅通过计算样本点的偏差来评估噪声水平。更改韵律可能会影响许多值，但音频样本听起来仍然很自然。例如，以 Lp 范数衡量时，愤怒时发出的言语可能与中立时发出的言语有很大不同，但它在语义上仍然有意义，并且听起来像人类。 因此采用并结合了 NISQA，这是一种最先进的基于 DNN 的语音评估系统，可以从 1 到 5 的范围内量化语音的整体质量和自然度。该术语经过加权并与 前面描述的对抗项形成完整的损失函数。

2.3.2 攻击说话人识别系统 SR

将目标 SR 系统表示为 SR(x)，它将音频 x 作为输入并返回识别的说话者标签。与 ASR 系统不同，说话者识别算法通常包含一个用于决策的阈值 θ 。对于托管 n 个注册说话者{s1,s2,...sn}的 SR 系统，可以将其建模为：

$$SR(x) = \begin{cases} \underset{s_i \in G}{\text{argmax}}\ S_i(x), & \text{if } \max\limits_{s_i \in G} S_i(x) \geq \theta \\ Reject, & \text{otherwise.} \end{cases}$$

其中 Si(x)是计算出的说话者 Si 的分数。因此，攻击目标可以表述为：

$$SR(x^*) = s_t, \text{s.t. } x^* = G(t_0, x_0, p), s_t \neq s_0, H(x^*) = H(x)$$

其中 s0 和 st 分别是原始说话者和目标说话者。对抗性样本可以通过最小化由对抗项 LSR(SR(x*),t)和质量评估项组成的损失函数来获得：

$$Q(x^*): x_{adv} = \text{argmin}_{x^*} L_{SR}(SR(x^*), t) + wQ(x^*)$$

虽然针对 ASR 和 SR 的攻击具有相似的问题表述，但这两个系统的运行方式不同，因此需要使用不同的攻击方法。SR 识别任务包含一个唯一的阈值 θ，只有当说话者 si 的得分 Si(x)在所有登记的说话者中最高并且也超过 θ 时，该说话者 si 才会被识别。 因此，对抗项可以写为：

$$L_{SR}(x) = \max\{\theta, \max_{s_i \in G \setminus \{t\}} S_i(x)\} - S_t(x)$$

考虑到攻击的黑盒设置，攻击者无法获得阈值 θ 。受数学中微分概念的启发，通过迭代缩小阈值范围来近似阈值。正如下图 5 中概述的完整算法所示，提出的阈值估计方案与对抗样本优化同时运行。

Algorithm 3: SMACK against black-box SR

Input: Initial range of threshold $[inf_0, sup_0]$
Target speaker s_t
Output: Adversarial audio example x_{adv}

pop $\leftarrow [p] *$ popSize
$\theta \leftarrow (inf_0 + sup_0)/2$
function S(pop)
  popAudio $\leftarrow G(t_0, x_0, \text{pop})$
  Get scores $S_i(x)$ for each $s_i \in G$
  return $\max_{s_i \in G} S_i(x)$
for i in range(N) **do**
  fitness $\leftarrow Fitness(\text{pop}, t, \theta)$
  for j in range(popSize) **do**
    if $S(\text{pop}) > \theta$ **and** $SR(\text{pop}[j]) =$ Reject **then**
      inf $\leftarrow S(\text{pop}[j])$
    else if $S(\text{pop}) < \theta$ **and** $SR(\text{pop}[j]) \neq$ Reject **then**
      sup $\leftarrow S(\text{pop}[j])$
    end if
  end for
  $\theta \leftarrow (inf + sup)/2$
  bestPop $\leftarrow$ pop[$Argmax$(fitness)]
  if $Fitness(\text{bestPop}, t, \theta) <$ Threshold **then**
    Generate children with *InsDel*, *Mutate*, and *Crossover*
  else
    for k in range($\mathcal{K}$) **do**
      $\theta \leftarrow \frac{\mathcal{K}-k}{\mathcal{K}}\text{inf} + \frac{k}{\mathcal{K}}\text{sup}$
      Estimate gradient and update bestPop
      if $S(\text{bestPop}) > \theta$ **and** attack fail **then break**
    end for
  end if
end for
get x_{adv} generated by $G(t_0, x_0, \text{bestPop})$
return x_{adv}

图 5 针对黑盒 SR 的 SMACK 算法

三、效果描述

本文对 SMACK 的实验效果进行了评估，测试了它对五个 ASR 系统和两个 SR 系统的攻击能力。

实验评估主要基于平均成功率（MSR）和词错误率（WER），这两项指标能够直接反映攻击的效果。WER 是衡量自动语音识别系统性能的一个重要指标，它表示识别结果与真实文本之间的差异。在本实验中，如果对抗样本的转录结果与目标短语一致，则被视为成功。此外，还记录了攻击过程中的查询数量和资源占用情况，以及使用 NISQA 分数来评估对抗样本的音频质量和自然度。

3.1 ASR 攻击结果

在 ASR 攻击实验中成果显著。实验结果如图 6 显示，五个不同的 ASR 模型平均 WER 为 11.42，平均成功率达到了 84.9%。这一成功率与训练有素的转录模型的表现相当，例如 DeepSpeech 2 的 WER 为 10.46。此外，对抗样本在音频质量方面表现出色，平均 NISQA 得分为 3.31，这一得分与正常人类语音的得分相近，表明我们的对抗样本在听觉上与正常语音无显著差异。

ASR system	WER	MSR	NISQA	Queries
CMU Sphinx	14.0%	79.7%	3.23	1386
DeepSpeech 2	9.8%	88.3%	3.30	1275
Google API	13.6%	81.8%	3.46	1149
iFlyTek	7.3%	90.6%	3.20	908
Azure	12.4%	84.2%	3.34	1067

图 6 ASR 实验结果

值得注意的是，成功实施攻击所需的平均查询次数为 1157，这一数字远低于传统基于 Lp 范数的对抗样本方法所需的 9620 次查询。这一显著的查询效率提升，本文认为可以归因于音素相似度在优化过程中提供的更有效的指导。

在资源消耗方面，算法在实验中占用了 2846 MB 的 GPU 内存，每次迭代耗时约 0.41 秒，而每个样本的平均处理时间为 474 秒。这些数据表明，尽管 SMACK 算法在处理复杂性上有所提高，但其资源消耗仍在可接受范围内。

针对恶意命令的攻击实验进一步证实了 SMACK 的有效性。实验结果显示，SMACK 在这些特定攻击场景下的平均成功率为 87.7%，这表明即使是针对特定的、

预设的恶意命令，SMACK 也能够保持较高的攻击成功率。

3.2 SR 攻击结果

在说话者识别（SR）系统的攻击实验中，SMACK 同样展现出了卓越的性能。实验设计涵盖了开集识别（OSI）、闭集识别（CSI）和说话者验证（SV）三种不同的任务类型，并且考虑了性别间和性别内两种不同的攻击场景。实验结果如图 7 显示，攻击的平均成功率为 99.2%。

Attack Type	Task	GMM-UBM			ivector-PLDA		
		MSR	NISQA	Queries	MSR	NISQA	Queries
Intra-gender	CSI	100%	3.42	645	100%	3.39	753
	OSI	100%	3.46	574	95%	3.27	966
	SV	100%	3.35	713	100%	3.41	978
Inter-gender	CSI	100%	3.44	879	100%	3.21	868
	OSI	100%	3.46	794	100%	3.32	1034
	SV	100%	3.31	671	95%	3.40	1309

图 7 SR 实验结果

尽管在某些情况下，攻击需要更多的查询次数来找到最优解，但这些情况并不常见。事实上，即便在性别差异较大的攻击场景中，攻击的可行性也并未受到影响，成功率依然达到了 99.2%。此外，对抗样本的 NISQA 平均值为 3.37，进一步证实了 SMACK 生成的样本在自然度和音频质量上的优势。

为了理解人类对语义对抗性音频的感知，本文还进行了用户可辨识性研究。在总共 168 名参与者中，大多数参与者认为语义音频样本的保真度和自然性优于传统的 Lp 范数对抗性攻击。

3.3 结论

实验不仅验证了 SMACK 的有效性和实用性，还展示了其在不同 ASR 和 SR 系统上的广泛适用性。实验结果清晰地表明，SMACK 能够在保持高音频质量和自然度的同时，有效地实施对抗性攻击。这些发现不仅为语音识别系统的安全性提供了新的视角，也为未来的研究和防御机制的开发提供了重要的参考。

四、评价与思考

SMACK 的研究为语音识别和说话者识别系统的安全性提供了新的视角。作为一种新颖的对抗性音频攻击手段，其核心优势在于通过仅操纵语义属性，能够在不降低语音质量的前提下，生成具有高隐蔽性的对抗样本。这一点与传统的基于噪声的对抗样本生成方法形成鲜明对比，后者往往因引入的噪声而留下明显的人工痕迹。SMACK 的攻击效果与现有对抗性音频攻击相似，目的都是误导 ASR 和 SR 系统，但它通过保持语音的自然性，提高了攻击的成功率和隐蔽性。这种攻击方式的提出，无疑为系统防御者带来了新的挑战，迫使他们重新考虑和设计更有效的安全防护措施。

此外，虽然 SMACK 生成的对抗样本与 DeepFake 在本质上有所不同，但两者在检测和防御策略上存在共通之处。DeepFake 检测通常涉及两个关键步骤：特征选择和分类器训练。然而，现有方法在泛化能力上存在局限，它们往往只能针对特定类型的数据集或欺骗技术有效。SMACK 通过操纵真实语音的韵律属性来生成对抗样本，这对 DeepFake 检测算法提出了新的挑战。

然而，SMACK 也有一些局限性。例如，它假设对抗性音频样本可以预先生成，并且整个对抗性音频样本可以作为目标系统的输入。这在实时在线攻击场景中可能不适用，攻击者需要实时生成对抗性音频扰动并与背景声音一起播放。此外，SMACK 专注于韵律的修改，对于与目标转录在音素上差异较大的原始语音，生成对抗性样本难度较大。

由于 SMACK 依赖于对韵律的恶意操纵，旨在检测基于 Lp 的对抗性音频样本中的人工痕迹的现有防御措施的有效性可能受到限制。未来潜在的防御和检测方向是通过对抗性训练策略将攻击引入训练管道中以生成新模型。为了实现这一目标，防御者需要访问模型和数据，并设计有效的白盒攻击以实现迭代对抗训练。另一个潜在的防御策略在于所谓的活性检测，其中可以利用人类语音的物理方面引起的声学特征来检测攻击。

总之，未来可以探索更多语义属性的对抗性操纵，并考虑实时攻击场景下的策略。同时，SMACK 的提出也促使安全研究者思考如何设计更为健壮的语音控制系统，以及如何改进现有的检测和防御机制，以抵御这类新型的语义对抗性攻击。

实验一：图像 LSB 隐写与分析方法

综合评分：

【实验目的】：

1. 熟悉并掌握 Matlab 基本操作；
2. 掌握 LSB 替换顺序嵌入隐写方法并实现；
3. 掌握 LSB 替换随机选取嵌入位置方法并实现；
4. 掌握卡方隐写分析方法并实现；

【实验内容】：

1. 学习 Matlab 基本操作；
2. 实现 LSB 顺序替换隐写与提取；
3. 实现 LSB 随机替换隐写与提取；
4. 实现卡方隐写分析；

【实验工具及平台】：

■ Windows+Matlab　　□ 其它：（请注明）_____

【实验涉及到的相关算法】：

1. LSB 顺序嵌入和提取

（1）嵌入过程：

读取图像并转换为 Double 型数据；
读取待嵌入的密文转化为二进制序列；
循环遍历像素点，顺序选取载体图像最低有效位替换为秘密信息；
生成载密图像保存；
绘制原始图像和隐写图像的局部直方图；

（2）提取过程：

读取载密图像并转换为 Double 型数据；
新建文件用于保存解密信息；
顺序提取前 n 个像素的最低有效位信息并保存；

2. LSB 随机嵌入和提取

与顺序隐写相比，随机隐写的差别主要在于循环遍历过程中需要增加随机选取像素点的算法。

（1）嵌入过程：

使用伪装密钥 k 作为伪随机数生成器的种子，生成伪随机数序列 k_i，嵌入位置 j_i 如下定义：

$$j_1 = k_1$$
$$j_i = j_{i-1} + k_i$$

（2）提取过程：

使用与嵌入过程同样的伪装密钥 k 作为伪随机数生成器的种子，生成的伪随机数序列一致，故可按照嵌入的位置依次选取最低有效位，从而解密。

随机隐写的嵌入和提取流程如下图 1 所示：

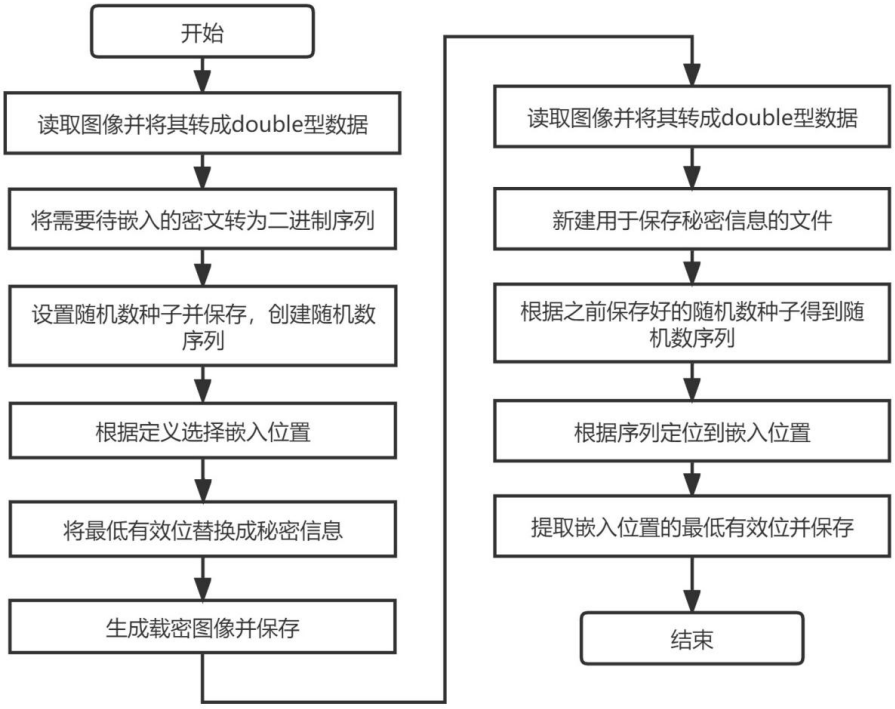

图 1 随机隐写嵌入与提取流程

3. 卡方隐写分析

嵌入信息会影响直方图的分布，由差别很大变成近似相等，但不会改变像素值为奇数和偶数的像素点总量，因为样值要么不改变，要么在 2i 和 2i+1 之间改变。

$$h^{*}_{2i} = \frac{h_{2i} + h_{2i+1}}{2}$$

即上述值在隐写前后不会发生改变，所以可以构造：

$$r = \sum_{i=1}^{k} \frac{(h_{2i} - h^{*}_{2i})^2}{h^{*}_{2i}}$$

其中，r 为卡方统计量，服从卡方分布，r 越小代表图片嵌入秘密信息的可能性越大。

最后再利用 matlab 自带的工具库计算图像服从卡方分布的概率：

$$p = 1 - \mathrm{chi2cdf}(r, k-1)$$

p 越接近 1，说明载体图像越可能携带秘密信息。

【实验分析】：

1. LSB 顺序嵌入和提取

绘制原始图像和嵌入图像以及它们各自的直方图，如图 2，可以看到嵌入后图像与原始图像肉眼几乎观察不到区别，但是观察直方图，可以发现有较大变化，出现了“值对”现象。

符合我们通过直方图分析可以看出一个图像是否被顺序嵌入隐写信息的结论。

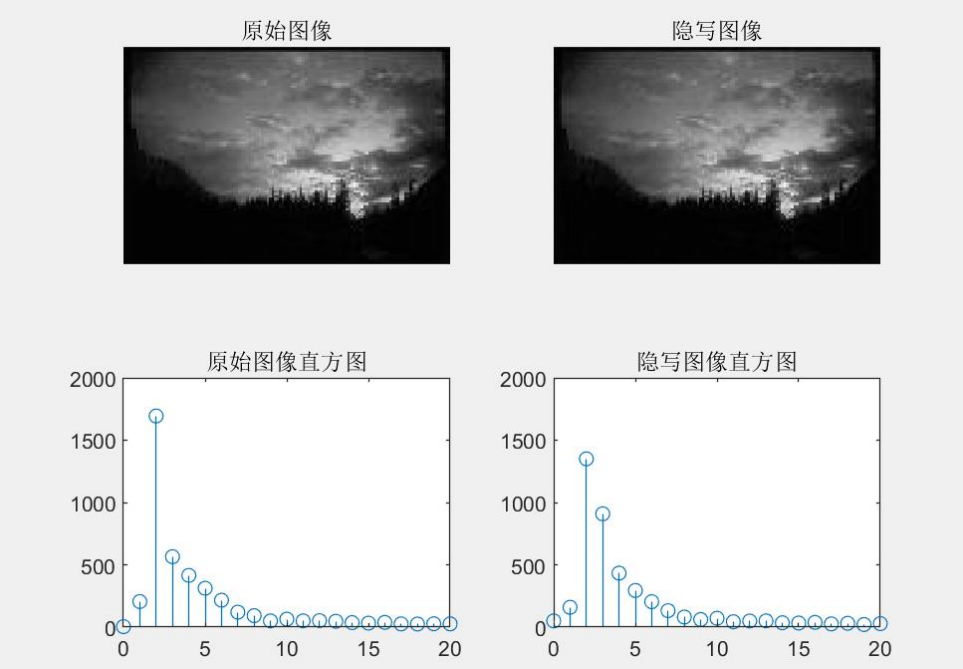

图 2 顺序隐写结果

2. LSB 随机嵌入和提取

如图 3 所示，可见嵌入信息前后的图像以肉眼观察几乎看不到区别。我们尝试比对两个图像的局部直方图，发现嵌入信息后的图像也没有明显的“值对”现象。这是正确的，因为随机隐写避免了载体图像的满嵌，这说明通过随机隐写可以确实可以在一定程度上绕过直方图检测的方式。

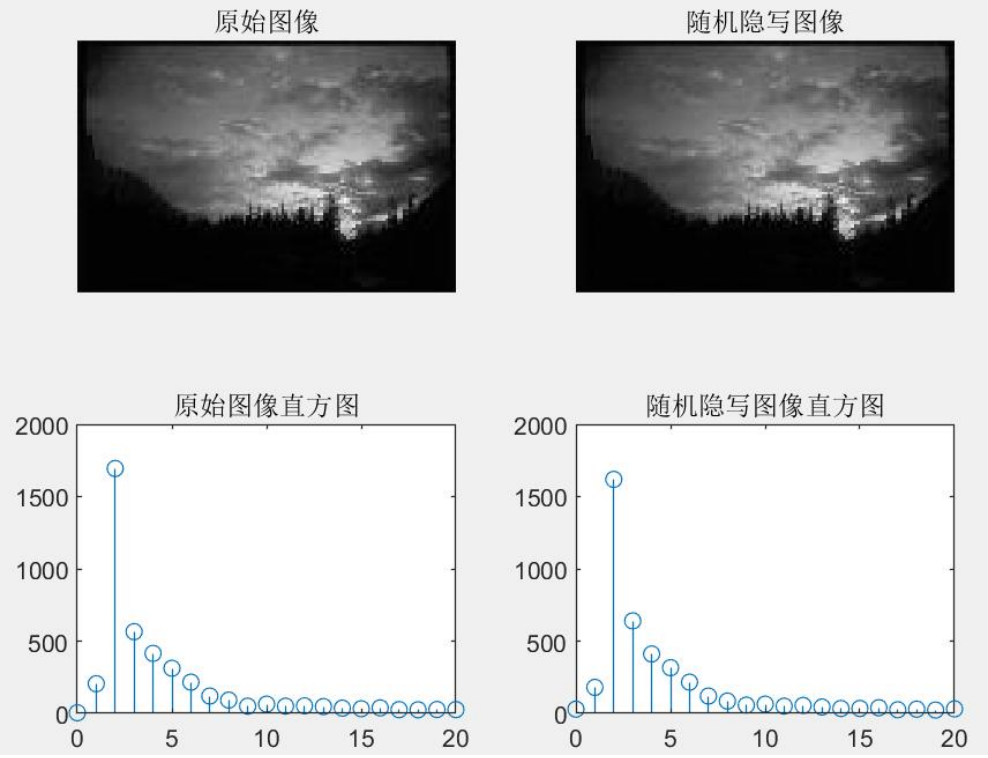

图 3 随机隐写结果

3. 卡方隐写分析

对顺序隐写且满嵌的图像进行卡方隐写分析，如图 4 其 p 值为 0.86908，已经是比较大的概率了，可以通过这个现象说明该图像中含有秘密信息。同时对比图 5 直方图分析也可观察到，值对现象较为明显，说明分析正确。

```
66    % 显示结果
67    disp(['卡方统计量对应的累积概率 p 值为: ', num2str(p_value)]);
```

命令行窗口

```
>> lsb3
卡方统计量对应的累积概率 p 值为: 0.86908
```

图 4 卡方分析结果

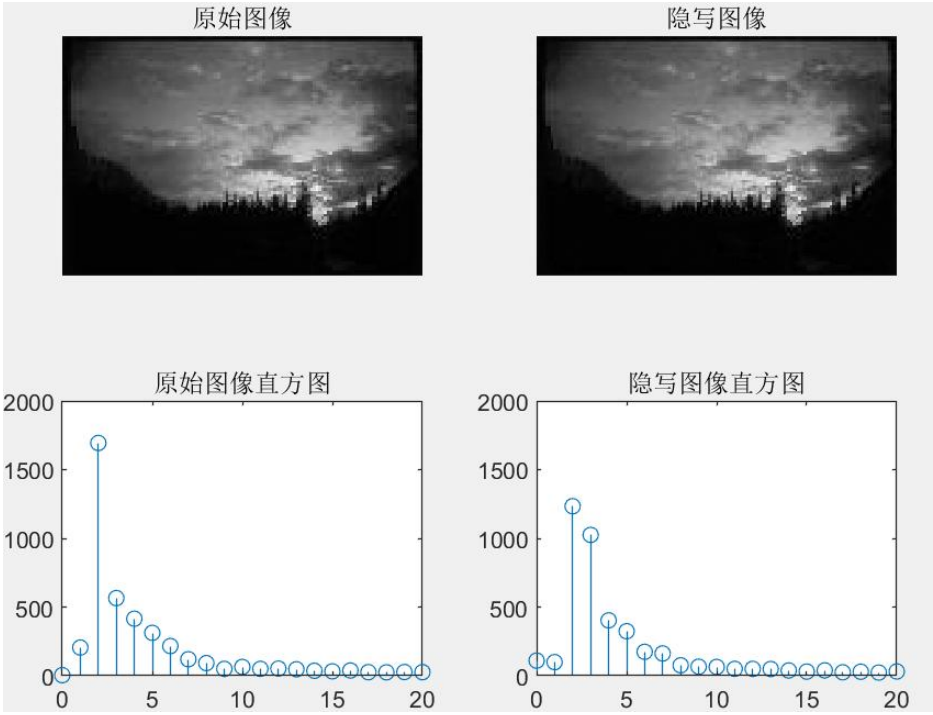

图 5 顺序满嵌结果

实验二：图像 JSTEG、F3、F4 隐写

综合评分：

【实验目的】：

1. 掌握 JSTEG 嵌入与提取隐写信息的方法并实现；
2. 掌握 F3 嵌入与提取隐写信息的方法并实现；
3. 掌握 F4 嵌入与提取隐写信息的方法并实现；

【实验内容】：

1. 实现 JSTEG 隐写与提取；
2. 实现 F3 隐写与提取；
3. 实现 F4 隐写与提取；

【实验工具及平台】：

■ Windows+Matlab　　□ 其它：（请注明）_____

【实验涉及到的相关算法】：

1. JSTEG 隐写与提取

（1）嵌入过程

首先对 JPEG 图片进行解码得到 DCT 系数，之后顺序将不为 0 和 1 的 AC 系数的最低比特位按照以下规则进行替换。最后再将 DCT 矩阵重新编码保存为 JEPG 格式即可。

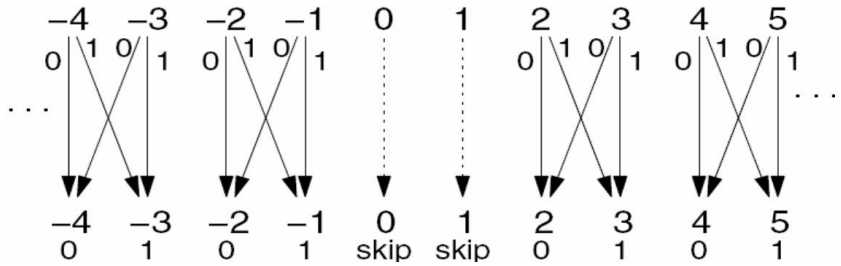

图 1 JPEG 隐写规则

（2）提取过程

首先从 JPEG 文件中提取出 AC 系数，之后顺序将不为 0 和 1 的系数按照上述规则进行加密信息读取：奇数为 1，偶数为 0。由此可得加密信息。

2. F3 隐写与提取

（1）嵌入过程

F3 的隐写过程类似 JSTEG，只需要将其替换规则更改如下图 2 即可。值得注意的是，如果原 DCT 系数为+1 或者-1，而待嵌入的比特位为 0，则先将其 DCT 系数改为 0，并重新选择嵌入位；如果原 DCT 系数为 0，则直接跳过，重新选择嵌入位。

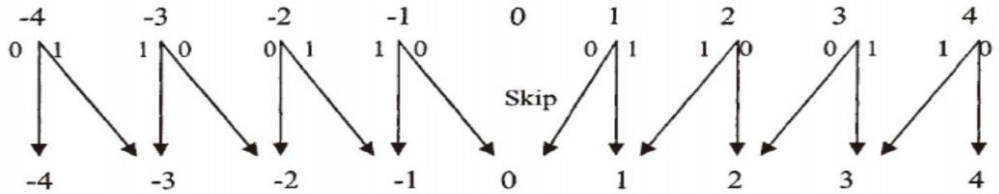

图 2 F3 隐写规则

（2）提取过程

F3 的提取过程也类似 JSTEG，只需将提取规则改为上述规则，当 AC 系数为 0，直接跳过；AC 系数为奇数，则得到密文比特为 1；AC 系数为偶数，则得到密文比特为 0。

3. F4 隐写与提取

（1）嵌入过程

F4 的隐写过程类似 JSTEG，只需要将其替换规则更改如下图即可。值得注意的是，如果原 DCT 系数为+1 且待嵌入的比特位为 0 或者 DCT 系数为-1 且待嵌入的比特位为 1，则先将其 DCT 系数改为 0，并重新选择嵌入位；如果原 DCT 系数为 0，则直接跳过，重新选择嵌入位。

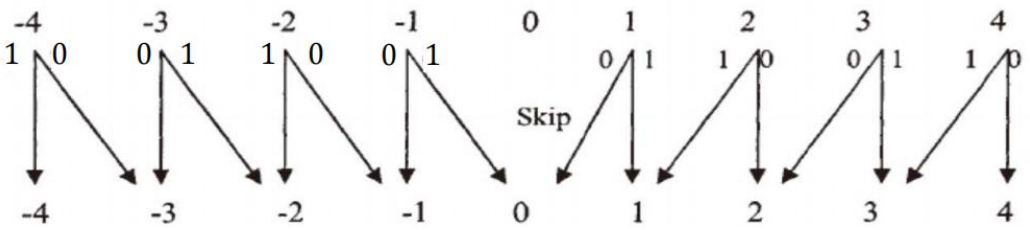

图 3 F4 隐写规则

（2）提取过程

F4 的提取过程也类似 JSTEG，只需将提取规则更改为：AC 系数大于 0，直接取出 LSB；AC 系数小于 0 时，取出 LSB，若 LSB 为 1 则密文比特为 0，若 LSB 为 0 则得到密文比特为 1。

【实验分析】：

1. JSTEG 隐写

对比原始图像和隐写图像的直方图如图 4，可见隐写前后的直方图变化符合隐写规则。

由于 JSTEG 对 DCT 为 1 和 0 的载体直接跳过不处理，所以隐写前后 DCT 直方图中 DCT 为 0 和 1 的方形不变；同时因为 JSTEG 用隐写后的载体 DCT 值的奇偶来表示隐写的比特是 1 还是 0，由此我们可以看到隐写后的直方图出现了一定程度的“值对”现象。

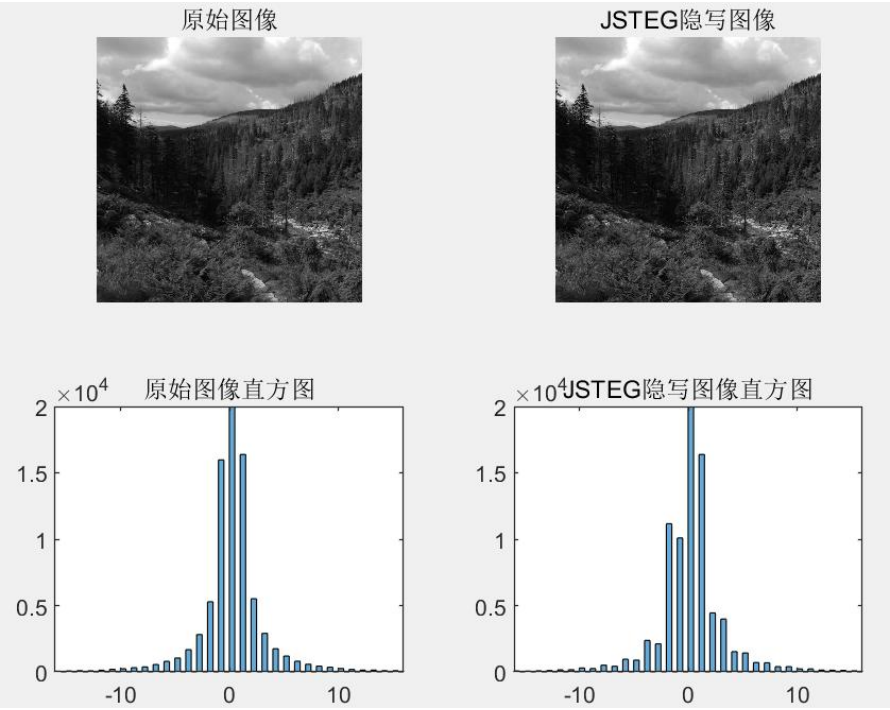

图 4 JSTEG 隐写结果

2. F3 隐写

绘制 F3 隐写前后的图像对比如图 5，可见变化符合 F3 的隐写规则。

由于 F3 隐写的时候，当原 DCT 系数为+1 或者-1，而待嵌入的比特位为 0 时会将其 DCT 系数改为 0，因此可以在隐写后的直方图上观察到 DCT 值为+1 与-1 的方形大幅缩短，而 DCT 值为 0 的方形增加。

根据 F3 隐写规则，不难得到 DCT 直方图偶数系数会增多，这也能从隐写后的直方图上观察出来。明显观察到跟隐写前相比，奇数系数的方形缩短了，而偶数系数的方形略有增加。导致两者差异减小，“值对”现象较为明显。

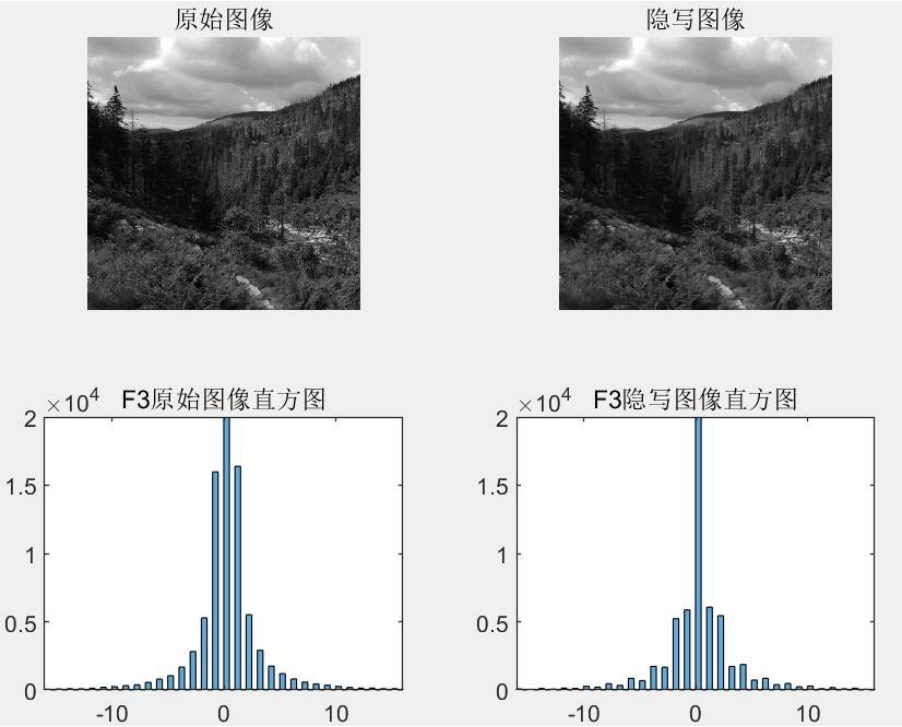

图 5 F3 隐写结果

3. F4 隐写

对比原始图像和隐写图像的直方图如图 6，可见隐写前后直方图变化符合 F4 隐写规则。

根据规则，隐写中 DCT 系数有一部分的-1 和 1 会变成 0，从而 0 的比例会增加，而-1 和 1 减少，可以从结果图看到确实如此。

根据 F4 的规则，不论嵌入的秘密比特位为 0 还是 1，都可能产生无效隐藏而需重新嵌入，重新嵌入的信息既可能是 0 也可能是 1，故可以得知前后的 DCT 系数直方图特性不会被破坏，并且从结果图中看到几乎没有“值对”现象，可知确实如此，符合规则。

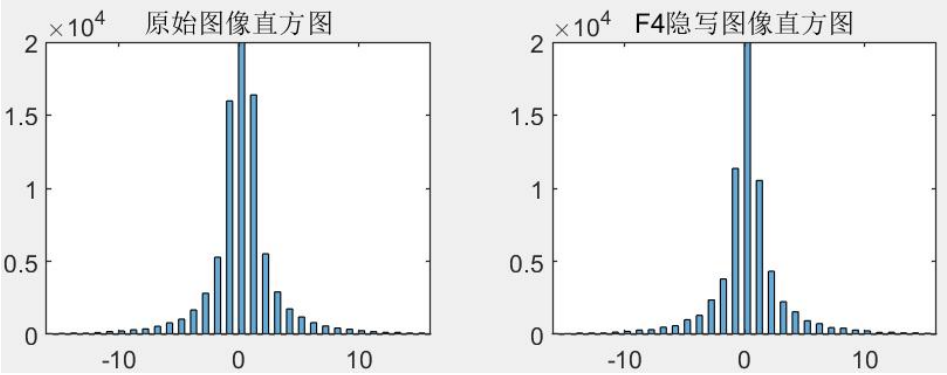

图 6 F4 隐写结果