武汉大学国家网络安全学院
课程项目报告

基于摩尔纹伪装的隐蔽抗摄屏水印

专 业 名 称 ： 信息安全

课 程 名 称 ： 信息隐藏技术

指 导 教 师 ： 任延珍

姓名	马锦贵	**学号**	2020302181036
姓名	王仕信	**学号**	2020302181112
姓名	熊银川	**学号**	2019302180049
姓名	周正业	**学号**	2020302181039

二〇二三 年 一 月

目录

摘要

摄屏场景的广泛应用和低廉的使用成本使得摄屏场景下的版权保护和泄密追溯成为学术界和工业界关注和要解决的问题。目前的研究工作通过增强传统的数字水印技术的鲁棒性来抵御摄屏场景下图像水印的形变失真，颜色失真，摩尔失真和高斯失真，现有抗摄屏水印的研究成果已经具备很高的鲁棒性，此外基于生成网络的抗摄屏水印在隐蔽性上取得突破性进展，生成的水印信息可见度逐渐变小。本作品立足于摄屏场景，利用摄屏场景下自然生成的摩尔纹现象，提出基于摩尔纹伪装的隐蔽抗摄屏水印，将水印信息以摩尔纹的形式嵌入图像，经过测试，摩尔纹实现的抗摄屏水印鲁棒性能够达到接近 100% 的提取准确率，比特错误率控制在 5% 以内。本项目设计了摩尔纹生成模块来对图像嵌入水印信息，水印提取网络采用数据增强与自监督学习的方式进行训练，并提出改进的 vote 方案-poss 来对网络输出层的节点进行冗余校验，辅以 Hamming 纠错码进行水印检错，以上方法使得水印提取具有很强的鲁棒性。同时，摩尔纹水印的隐蔽性不是来源于视觉上的不可见，而是来源于视觉感知的自然性，尤其是经过了摄屏操作，摄屏的失真对视觉的影响已经足以掩盖水印摩尔纹的突兀性，因此视觉感知很小，具备很强的隐蔽性。

关键词 抗摄屏水印 摩尔纹 深度神经网络

一、 作品概述

1.1 研究背景

保密工作是许多党政、军事、工商业保护信息安全的重要需求，其中泄密追溯是做好保密工作的关键一环，传统方法上采用鲁棒数字水印技术对涉密文件添加水印能够做到对泄密文件的来源追溯。但是，随着智能手机和摄像机的广泛普及，摄屏技术逐渐成为一种方便快捷、成本低廉，高效率的信息传输形式。使用手机或摄像机摄屏进行窃密，成为违法分子惯用的方式。摄屏后的图像能够保留屏幕上大部分的关键主体信息，但是在视觉质量上会发生较大程度的失真，导致传统的数字水印在摄屏后的图像上没能保存下来。因此，研究一种能够有效抵御摄屏带来的图像失真的抗摄屏水印技术，是解决摄屏泄密追溯的关键问题。

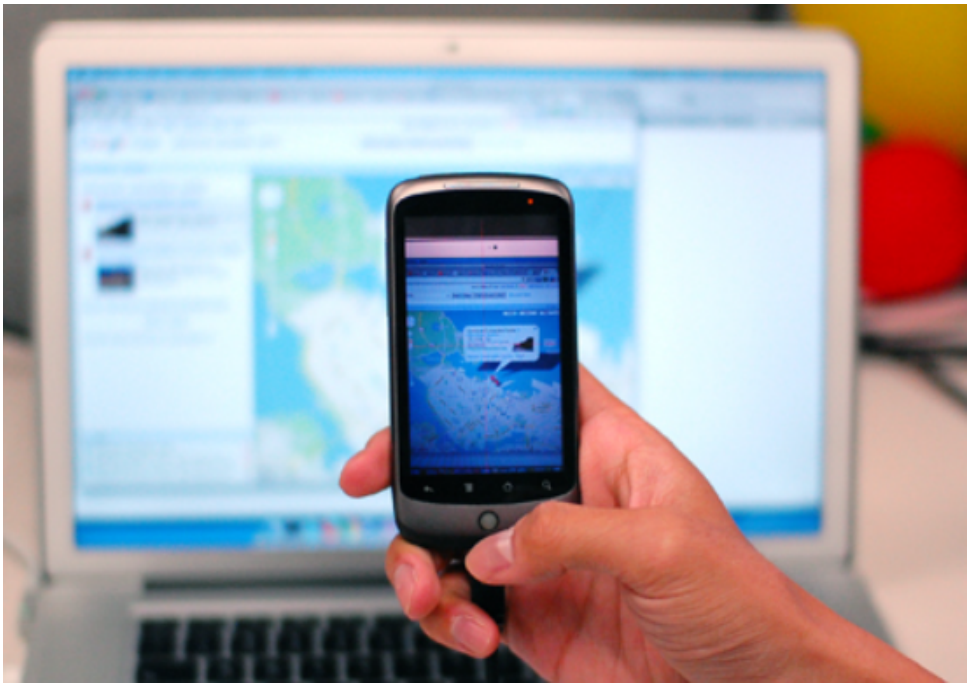

图 1: 摄屏技术的广泛使用

与传统数字水印技术面临的压缩、剪裁和模糊等失真问题不同，抗摄屏水印需要充分考虑摄屏场景中特有的主要失真因素，文献 [2] 将其归纳为：拍摄角度引起的几何形状失真，光照因素引起的颜色亮度的失真，光衍射带来的摩尔纹失真以及其它环境和设备带来的干扰因子（我们将其归结为高斯失真）。与传统数字水印技术索要考虑的问题原理近似，抗摄屏首先要解决的是水印鲁棒性的问题，抗摄屏技术的鲁棒性要求水印能够抵御摄屏过程带来的多种失真因素并在水印提取一方能够有效提取。其次是水印的隐蔽性问题，原则上水印在视觉上不得对原始图像的主体信息有太大程度的破坏，在保密工作等一些重点领域，这个要求被强化，要求水印在原始图像载体上要足够隐蔽，隐蔽包括两个层面的内容，第一是水印在视觉上的感知要尽可能地小，即不被看见，第二是水印存在于载体中对人而言要尽可能显得自然，即不被发现。最后是一些其它的小问题如水印嵌入信息容量的问题、水印效率问题等。

图 2: 摄屏过程对图像带来的失真

为了满足水印隐蔽性需求，现有的抗摄屏图像水印算法主要是通过寻找变换不变域，将水印信息嵌入到不变域内，以期望抵抗屏幕拍摄带来的一系列图像失真，如亮度失真，对比度失真，饱和度失真和图像压缩等失真。通过寻找变换不变域方法来嵌入水印，图像质量取得了一定的效果。然而，这些算法并未考虑图像纹理简单的特点，水印信息易受摄屏过程影响，提取精度较低，并不能适用于现实场景。此外，现有的图像水印嵌入方法往往在图像被其他设备拍摄后就会产生严重失真，很难在屏摄图像上提取或检验出水印，因此需要一种能够抵御摄屏攻击，同时兼顾图像质量的水印嵌入方法。

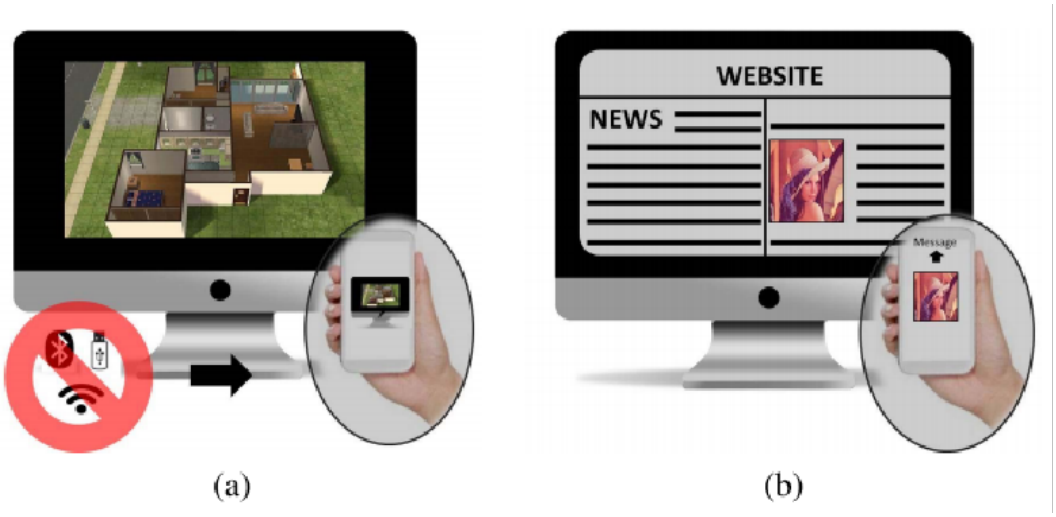

图 3: 抗摄屏技术的应用

本项目考虑到抗摄屏水印鲁棒性和隐蔽性之间的矛盾关系，提出以摩尔纹这一摄屏场景下天然存在现象作为抗摄屏水印信息表达的载体，通过将水印信息伪装成摩尔纹现象嵌入图像并进行提取，能够有效兼顾抗摄屏水印对鲁棒性和隐蔽性的需求。

1.2 相关工作

抗摄屏技术是信息隐藏和数字水印领域近年来的研究热点，2018 年，Han 在 screen-shooting resilient watermarking [5] 文章中提出了一种抗摄屏水印算法，利用强度的尺度不变特征变换算法来定位特征区域，并在相关区域的离散余弦系数 (dct) 中进行水印的嵌入和提取。该方法在应用的过程中给水印图像加上可见的特制边框，便于在屏幕拍摄时准确地定位到目标图像的位置信息，进行透视矫正，虽然具有较好的抗摄屏性能，但是该方法采用的加特制边框操作，在一定程度上影响了图像的美观而限制了算法的应用，并且边框容易遭受破坏，使得后续的水印提取失败。该文章是抗摄屏水印的开山之作。PIMOG[1] 引入了名为 PIMoG 的屏幕拍摄噪声层这一概念，通过定量屏幕拍摄过程的最大失真来生成具有强鲁棒性的效应噪声层。该论文将最大失真归纳为三个部分：透视失真、照明失真和摩尔纹失真，利用噪声层对整个网络进行端对端训练。并通过大量的实验数据证明该 PIMoG 噪声层具有优越的性能。同时该论文还提出了梯度掩模引导的图像丢失和边缘掩模引导图像丢失，进一步提高了整个网络的鲁棒性和不可见性。TERA[2] 讨论了“屏幕到摄像机信息传输”的重要用户场景，发现现有的方法都不能同时满足四个重要特性，即高透明度、高嵌入效率、强传输鲁棒性和对设备类型的高适应性。这主要是因为这些属性相互矛盾。因此，该论文提出了一种称为“TERA”（透明度、效率、鲁棒性和适应性）的屏幕到相机图像代码，这使得首次可以避免上述四个属性之间的矛盾。通常，TERA 采用颜色分解原理来确保视觉质量，并采用基于叠加的方案来确保嵌入效率。进一步设计了基于 BCH 编码的信息排列和强大的注意力引导信息解码网络，以保证鲁棒性和适应性。通过大量实验，证明了该方法的优越性和广泛应用。Unseencode[3] 中，Hao Cui 等设计了一种支持可靠的基于图像的提取，并媲美传统条形码的隐形条形码 UnsenCode。该方案在嵌入时使用基于 VLC 的方法中的帧间其纳入来实现条形码的隐藏，在提取时使用基于彩色图像交叉分量相关性的方法来保证条形码的可靠性和有效性。并通过评估确认了 UnsenCode 的适用性和可靠性。另外，Han 还在 camera shooting resilient watermarking scheme for underpainting documents

中基于 CSR 水印算法，提出了一种基于打底的相机拍摄弹性水标记方案，并在打底上嵌入水印。通过在 DCT 域交换系数的方法嵌入水印以满足相机拍摄过程的鲁棒性和水印的隐蔽性；利用行间距区域和嵌入块的对称性，可以大大减少文本区域可能产生的噪声和失真；同时，设计了基于翻转的方法，以使打底具有强烈的对称性，进而增强由屏幕到相机的强传递鲁棒性。

下面的表格中总结了上述工作的主要思路和优缺点。

表 1: 相关工作对比分析

论文工作	创新点	优点	不足
Screen-shooting resilient watermarking[5]	基于强度的尺度不变特征变换（I-SIFT）算法	提取效率较高,鲁棒性较强	拍摄角度倾斜和受光照影响时鲁棒性大幅下降
PIMoG[1]	设计新噪声层和梯度掩模和边缘掩模引导图像损失两种损失训练方法	高隐蔽性和强鲁棒性	水印网络需要全幅图像的信息,图像轻微残缺将严重影响水印提取
TERA[2]	基于图像颜色分解,将信息隐藏在互补的两帧图像中	高透明度、高嵌入效率、强鲁棒性和对设备高适应性	一旦没有连续帧,单张图片的水印相比其它工作较明显,容易被发现
Unseencode[3]	设计了一种支持可靠的基于图像的提取的隐形条形码	可靠性和适应性较强	水印比较明显,图片显得不自然,容易被发现
Underpainting documents[4]	基于打底的相机拍摄弹性水标记方案,并在打底上嵌入水印	在 DCT 域嵌入水印,效果鲁棒性很强	对剪切比较敏感,依赖图像全局信息

1.3 项目特色

本项目的的核心思路是将水印信息伪装成图像摄屏自然产生的摩尔纹进行嵌入，总结起来本项目主要有三点特色。

1. 本项目立足摄屏场景，抓住摄屏过程的自然产物摩尔纹作为切入点，将水印伪装成摩尔纹嵌入图像，水印显得自然隐蔽。

2. 本项目引入自监督学习训练的深度神经网络作为水印提取器，辅以几何矫正、冗余嵌入和 Hamming 纠错码技术，有效抵御摄屏中的各种失真，鲁棒性强。

3. 本项目代码模块划分清晰，具有很强的扩展性，支持自主重训练网络、数据集替换，自主设计水印，选择校验方式等拓展功能。

1.4 应用前景

本项目实现的抗摄屏水印技术能够有效弥补传统数字水印技术在摄屏场景下暴露出来的水印鲁棒性，水印隐蔽性的问题。该项目的技术成果有望在党政军和商业的保密工作中发挥重要的作用，具体的应用方向有泄密追溯、版权保护以及秘密通信。

泄密追溯。本项目设计并实现的基于摩尔纹伪装的抗摄屏水印技术能够为用于内部传播的投影幻灯片、涉密电子显示仪器上的图片等进行水印嵌入，即使涉密屏幕上的图像被不法分子摄屏处理，水印依旧能够鲁棒地保持，能够在第一时间有效提取出摄屏图像上的水印信息，从而精准定位泄密节点并对相关人员进行问责。

图 4: 泄密追溯

版权保护。对于具有高保密性的重要可视化作品用于屏幕展示时，可以使用本作品设计的摩尔纹抗摄屏水印对屏幕图像进行水印嵌入，通过在水印信息中记录版权问题等相关信息，能够在摄屏后的影像中通过水印提取网络获取水印中的版权信息，从而实施严格的版权控制。

图 5: 版权保护

秘密通信。抗摄屏水印本质上是一项信息隐藏技术，秘密通信的双方通过将秘密信息以摩尔纹的形式嵌入到图像中，通过拍摄图像后将拍摄图像在公共信道上进行传输信息接收方通过水印提取网络将秘密消息提取出来恢复明文，由于信息隐藏在摄屏图像的摩尔纹中，攻击者或第三方难以察觉到摄屏图像上的秘密信息，也无法将其提取出来，从而能够有效实现通信双方信息传输的隐蔽性和安全性。

图 6: 秘密通信

二、 作品设计与实现

2.1 设计目标

摄屏领域的信息隐藏在这个人人都有智能手机的时代具有重大意义，在泄密追踪，版权保护，亦或者单纯的传输信息等方面有广泛应用。摄屏不同于传统网络信道，除开编码本身可能产生的错误，摄屏还会带来诸如透镜畸变，光畸变，几何畸变，摩尔纹等强干扰，并且摄屏的使用场景大部分时候会使得图片处于倾斜状态，这一系列问题需要对编码进行精心设计，同时需要对图像进行一系列处理。我们将水印信息伪装成摄屏过程产生的摩尔纹，分块嵌入载体图片，待提取时，对图片进行预处理，随后用神经网络提取隐藏信息，实现摄屏水印。我们的作品有以下设计目标。

1. 水印的鲁棒性。抗摄屏水印需要能够抵御摄屏过程带来的各种失真，使得水印在经历摄屏后在摄屏图像上保留水印信息。2. 水印的隐蔽性。抗摄屏水印本身不能对图像的视觉效果带来较大程度的损害，原则上应该做到尽可能视觉上不可感知以及不可发现。3. 可接受的嵌入容量和嵌入效率。抗摄屏水印的嵌入率和嵌入效率应该控制在现实应用条件下可满足的范围内，才能适应广泛的抗摄屏水印的应用场景。

2.2 系统整体架构

本系统按照工作流程可以划分为两大部分，水印的生成嵌入和水印的提取。水印的生成嵌入主要包括摩尔纹的生成，水印校验序列的生成和水印的嵌入，水印嵌入可以使用下面的流程图来表达。

图 7: 水印嵌入流程

水印图片经过摄屏后，会发生形变导致的几何失真，光照导致的亮度失真和颜色失真，摩尔纹导致的摩尔失真以及其它失真（高斯失真）。摄屏后图片进行水印提取，提取的流程主要包括几何失真的矫正，水印提取网络的水印提取，水印序列的校验。水印的提取可以通过下面的流程图进行表示。

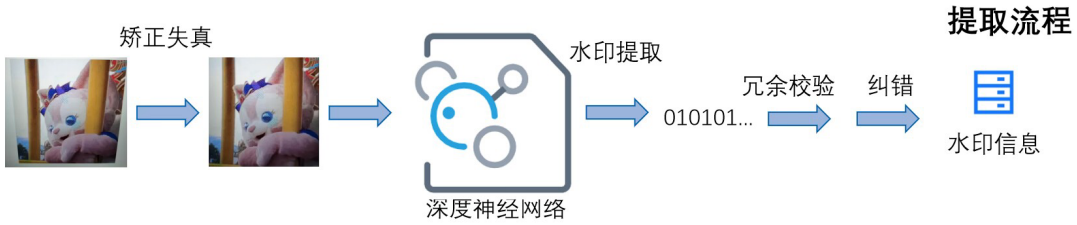

图 8: 水印提取流程

除了水印的嵌入与提取，本项目需要自己完成水印提取网络的训练和优化，其中包括数据集的获取，自监督学习训练，微调操作以及冗余校验 poss 方案。下面的流程图表达了这个过程。

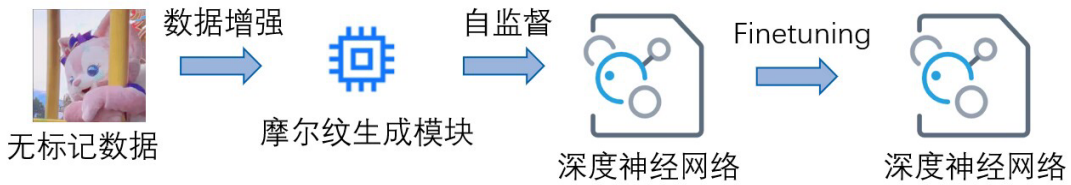

图 9: 水印提取网络训练流程

综上，本项目的主要工作可以包括以下若干模块的设计和实现。

图 10: 本项目主要设计与实现的功能模块

2.3 自适应隐蔽摩尔纹水印的生成与嵌入

2.3.1 摩尔纹的光学原理

对于摩尔纹，大英百科全书给出的定义是用等宽黑边条构成的光栅叠加在另一个相同的光栅上，以 1ř 到 45ř 的角度发生交叉产生的效果；维基百科给出的定义是一种大规模干涉图案，由具有透明间隙不透明条纹的图案经过微小的移位、旋转得到的新图案与原图案叠加生成；百度百科给出的定义未两条线/物体之间以恒定的角度频率发生干涉的视觉效果。它是光栅唯一精密测量的基础，具有极大的科学和工程价值，广泛应用在程控、数控机床和三坐标测量机、精密测量与定位、超精密加工、微电子 IC 制造、地震预测、质量检测、纳米材料、机器人、MEMS、振动检测等众多领域。但是，摩尔纹却对数字音视频的跨设备光学传输、版权信息的传递产生极强的干扰。

数码相机使用纵横排列的光照传感器阵列与彩色滤波片结合，将接收到的光照强度、颜色（红绿蓝）信息转换为电信号，再通过一系列处理得到数字信号存储 [23]。再由电信号到数字信号的处理过程中，彩色去马赛克时，对高频信号进行采样，大量的高频信息出现混淆；同时，Bayer 阵列的 RGB 分量交替排列且分布不均匀（绿：红：蓝 =2：1：1），导致产生摩尔纹 [24]。总而言之，感光元件 CCD（或 CMOS）像素的空间频率与影像中（LED）条纹的空间频率接近，就会产生摩尔纹 [25]。

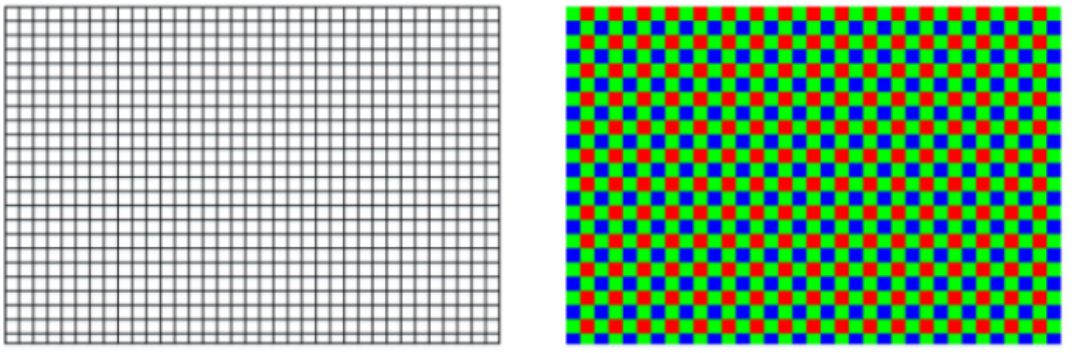

图 11: 相机感光阵列（左）和拜尔模式阵列（右）[23]

[26] 给出了摩尔纹的简化数学模型：

$$I(x,y) = A(x,y) + B(x,y)\cos[2\Pi(f_1 - f_2) + \Phi_o - \Phi_i]$$

其中，A(x, y)、B(x, y)分别为倾斜因子和调制因子，Φ_o、Φ_i分别为两幅初始条纹的初始相位。对摩尔纹的研究及生成具有一定的参考价值。

图 12: 摩尔纹简化数学模型

2.3.2 自适应摩尔纹设计原理

本文的摩尔纹是基于图像特效——摩尔纹 moir[27] 这一篇博客进行设计的。通过对图像中的像素点进行移位、旋转等操作，使图像出现摩尔纹效果。其步骤大体如下：

1) 建立参照点：选取图像所在平面上的一个点（相对与图像左上角像素点的位置，根据 matlab 的索引习惯，将该处的像素位置设为（1，1））作为坐标原点，并建立平面直角坐标系；

2) 取点并计算位置信息：依次选取像素点（算法中按由左到右、由上到下的顺序进行选取）作为目标操作像素点，计算出其弧度 beta(该点到坐标原点的连线与正 x 轴形成的夹角)、半径 radius(连线长度)；

3) 旋转：在上一步中得到的弧度 beta 的基础上，再加上半径乘以一个系数 degree（该值越大，生成的摩尔条纹曲折系数越大，需要根据图像的大小来调整以生成更加自然的摩尔纹），得到新点（源操作像素点）的弧度。

4) 取值：使用源操作像素点的像素值来替换目标操作点的像素值。一张彩色图拥有三个颜色通道，只需取其中一个通道执行上述操作，便能生成很好的摩尔纹效果（在嵌入摩尔纹时，我们选用的是第一个通道——红色通道）。如果对多个通道执行该操作，一方面会降低嵌入效率，另一方面会大大降低图像的质量。

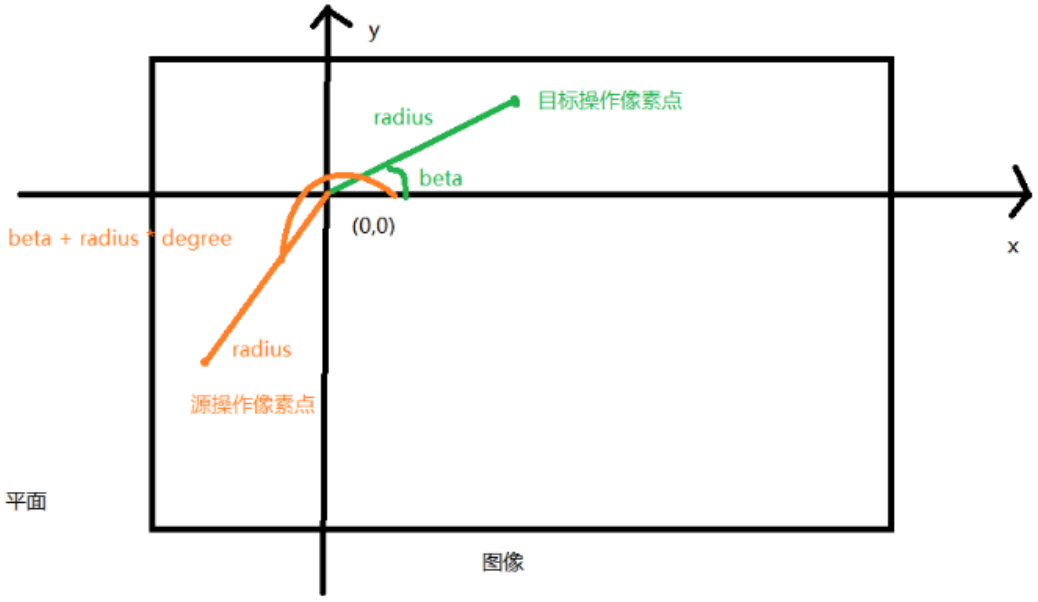

图 13: 摩尔纹生成原理

2.3.3 摩尔纹水印的嵌入

我们的摩尔纹并非是通过两张图像叠加，而是使用一个算法直接在原图像的基础之上进行编辑来生成的。因此，嵌入信息时，通过将不同的参数（坐标原点的位置）与不同的信息进行绑定，然后使用 2.3.2

中算法直接编辑原图像生成摩尔纹效果，而不是生成一个水印图像，然后再作矩阵加法将水印添加到原图像上。例如，我们在实验中，嵌入 1 时，我们将坐标原点设置为图像左上角像素点所在的位置；嵌入 0 时，我们将坐标原点设置为图像右下角像素点所在的位置。

我们结合《信息隐藏》这一门课上学习的 DCT 等隐写算法，选择图像分块作为 1 位信息的载体单元。这样保证了图像的嵌入信息量。同时，为了让生成的水印尽可能的显得自然，我们适当扩展了分块的大小，使用 64*64 大小的块来存储信息。嵌入算法伪代码如下。

Algorithm 1 The embedding algorithm for moire watermark

Input: block: an area to be embeded, info: the message of watermark, Degree: a controlable number of moire watermark which decides the strenth, Degree = 10

Output: output image block

get the block size (row,col)
set the origin point O
if info = 1 **then**
  $(Center_Y, Center_X) = (1, 1)$
else
  $(Center_Y, Center_X) = (row, col)$
end if
$out = block$
edit block
for all i *from* 1 *to row* **do**
  for all j *from* 1 *to col* **do**
    count the position (y1,x1) of pixel(i,j)
    $y1 = i - Center_Y$
    $x1 = j - Center_X$
    $$\text{beta} = \begin{cases} \text{atan}\left(\frac{y_1}{x_1}\right), & x_1 ! = 0 \\ \frac{\Pi}{2}, & x_1 = 0 \end{cases}$$
    $$\text{radius} = \sqrt[2]{x_1^2 + y_1^2}$$
    $beta\_tmp = beta + Degree * radius$
    $y_2 = radius * sin(beta\_tmp)$
    $x_2 = radius * cos(beta\_tmp)$
    $is = Center_Y + y_2$
    $js = Center_X + x_2$
    if (is,js) is in image **then**
      $out(i, j) = block(is, js)$
    end if
  end for
end for
return output image block

2.4 基于 Alexnet 的水印提取网络设计

2.4.1 改进的 Alexnet 架构

我们的水印提取环节是利用了深度神经网络强大的信息提取能力来完成的，一开始我们基于简单的 CNN 模型对每一个图像分块进行信息提取，但是效果并不好，最终我们通过对深度神经网络架构设计的大量文献调研，选择了针对于开源数据集 Imagenet 做图像分类的多分类神经网络 Alexnet[6]，Alexnet 的网络架构如下所示，

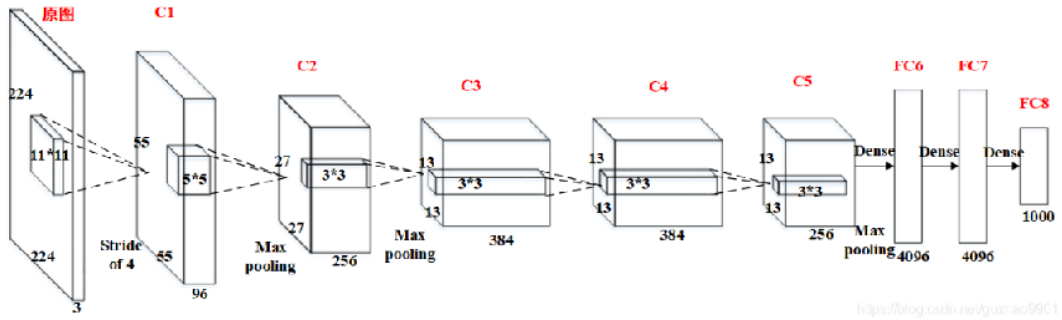

图 14: 原始 Alexnet 网络架构

该网络主要由 5 个卷积层和池化层加上 3 个全连接层构成，5 个卷积层用于提取图像的局部特征信息，在最后的一层卷积上已经捕获到了图像的大部分局部特征信息，将最后一层卷积层的激活值通过三个全连接层（线性层）进行特征的组合，并在输出层上输出每一个图像类别的置信度。

2.4.2 水印提取网络设计与细节理解

我们考虑到我们的数据集并没有 Imagenet 数据集庞大，如果使用原始的 Alexnet 将会导致没有足够的数据去训练网络的大规模参数，这样训练出来的网络很有可能会出现对训练数据过拟合而在测试集上表现较差的问题。同时我们水印信息是以比特为单位嵌入到图像的一个分块中，因此我们只需要输出层输出比特 0 和 1 的两个置信度值即可。考虑到这两个实际情况，我们做出以下修改。

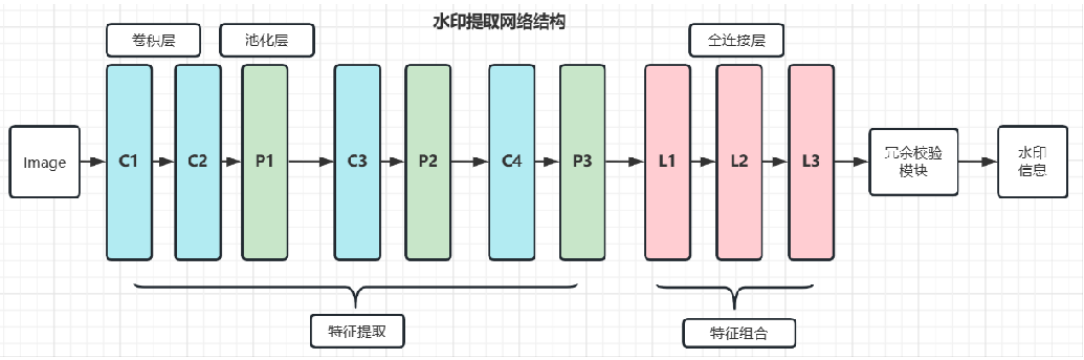

图 15: 改进的 Alexnet 网络架构

新的网络架构将特征提取的网络层修改为四个卷积层和三个池化层，特征组合和输出置信度的网络层保持三个全连接层不变，但是修改了每一层的输入和输出通道数，并保持输出层的激活单元为 2，分别对应 0，1 比特的置信度值。其中最接近输入层的是两个卷积层和一个池化层，后面都是一个卷积一个池化层，这样考虑是因为池化的过程会丢失一定的信息，因此我们希望前两层卷积层能够专注于图像局部信息的充分提取，而不要立刻进行池化。输入的图像经过水印提取网络后，进入冗余校验环节，最终输出提取出来的水印信息。

2.4.3 关键代码解读

我们基于 torch 库定义了水印提取网络相关的细节，由于我们使用的是三个通道 RGB 的图片，所以我们输入层是 3 个通道输入，输入层是 2 个激活单元。卷积核统一使用 5*5，卷积的步长为 1，使用 2 为边界填充大小，不使用其它参数，由于我们的网络结构相对比较简单，因此不使用 dropout 剪枝而是后面在训练的时候使用 finetuning 来进行模型的微调。

```
self.model = nn.Sequential(
      nn.Conv2d(3, 32, 5, 1, 2),#卷积层
```

```
        nn.Conv2d(32, 32, 5, 1, 2),#卷积层
        nn.MaxPool2d(2),#最大化池化层
        nn.Conv2d(32, 32, 5, 1, 2),#卷积层
        nn.MaxPool2d(2),#最大化池化层
        nn.Conv2d(32, 64, 5, 1, 2),#卷积层
        nn.MaxPool2d(2),#最大化池化层
        nn.Flatten(),#将特征扁平化处理
        nn.Linear(4096, 256),#全连接层
        nn.Linear(256,16),#全连接层
        nn.Linear(16,2)#全连接层
    )
```

2.5 基于自监督学习的水印提取网络训练与优化

2.5.1 基于数据增强技术的数据集获取

我们对摩尔纹的相关内容和可用资源进行了广泛的调研，但是基本上找不到与摩尔纹有关的神经网络资料以及开源的数据集，因此，我们只能自己构造数据集。我们选用了一批网络上免费可用的图片作为基础图片，使用我们的摩尔纹生成器在图片上生成表达水印信息的摩尔纹作为图像数据集。但是我们发现这种人工生成的方式不仅耗时较长，而且对生成的图像进行人工标注也是一件很复杂的事情。因此我们能够生成的数据集比较有限，经过长时间的生成，一共产生 2000 张有水印标注的摩尔纹图像数据集。但是这样的数据集除去测试集和微调数据集，要将网络进行充分的训练，强化模型的泛化能力还是不太理想。经过调研，我们采用 CV 领域常用的数据集扩充方法-数据增强。

数据增强也叫数据扩增，意思是在不实质性的增加数据的情况下，让有限的数据产生等价于更多数据的价值。它采用一些特定的技术来使得一张图像产生多个图像而不影响图像主要信息的存在。目前主要的数据增强技术包括图像翻转，图像裁剪，图像缩放，图像旋转等等。

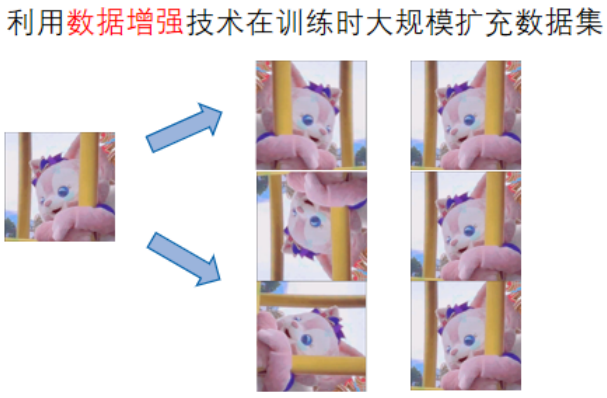

图 16: 数据增强

由于我们的水印图像要求对图像的完整性有一定的保护，所以我们不使用图像剪裁，另外图像的缩放会带来神经网络输入层的维度不匹配，所以我们最终经过多次实验测试，发现图像的翻转和颜色变化对训练模型的效果最好，因此我们选择了这两种数据增强方法进行数据集的扩充。

另外，值得一提的是，由于卷积神经网络的特点，卷积层善于捕获图像的局部特征，例如图像的颜色，形状等，但是我们希望神经网络去学习的不是图像主体的特征，而是嵌在上面的摩尔纹信息，但是摩尔纹信息相对于图像主体信息相对较弱，这样带来的后果有两个，第一是模型中的大部分参数学习到的可能是图像主体信息的特征而不是摩尔纹的特征，这就导致了大量神经元功能的浪费，第二是，这些记忆图像主体信息的神经元被激活后，不仅于我们的水印提取任务没有关系，还会干扰我们信息提取的准确率，一个形象的例子是，对于没有嵌入水印的图像，图像将其提取出的不同水印信息的概率是不一

样的，这说明训练数据集中图像的主体特征对摩尔纹与信息的映射关系造成了影响。因此我们做了下面这样的一个数据集增强方案。

我们将同一张基础图像同时嵌入不同水印信息的摩尔纹，得到多个水印图像副本，这些副本的主体信息是一样的，但是上面摩尔纹所代表的水印信息是不一样的，这有点类似对比学习和注意力机制的思想。我们将这样的数据投入训练，让模型能够学会忽略图像的主体信息，去关注学习图像上的摩尔纹与水印信息的对应关系，这样模型在预测时不会受到新数据集主体信息的影响，重点关注图像上的摩尔纹信息。

图 17: 让模型关注摩尔纹而非图像主体

2.5.2 水印网络的自监督学习训练

在机器学习领域，监督学习是指在有标注的数据上进行有师学习的一种学习方式，学习过程中，机器能够通过数据上预先标注好的监督信息或指导信息进行纠错和优化，无监督学习则不同，它是让机器在大量的没有标注的数据集上进行学习，自动获得数据集上的信息，监督学习需要提前获得标注数据集，这往往是一件昂贵的事情，无监督学习无需标注数据，但是学习效果往往不如监督学习好，而且需要根据特殊场景去设计学习策略。自监督学习是一种将无监督学习转化为监督学习的方式，它通过将无标注的数据集通过某种数据增强或者数据转化的方式，转化为另一批数据样本，同时生成对应数据变化的标签，这样就相当于无标注的数据进行了某种意义的标注。自监督学习本质是让机器自己添加数据集的标签并自己基于这些标签去学习数据集中的信息，这种技术通常使用在自动编码器中，用于提取图像中的特征信息，自监督学习训练得到的模型往往只是一个特征提取器，并不能直接作用于下游任务，必须在自监督学习网络的后端连接几层全连接层进行特征的组合并用监督学习的方法进行微调操作，这种方式在迁移学习中也有类似的应用。

在本项目中，为了解决没有数据集的问题，我们制作了一个摩尔纹生成器，如前文所说，它能够将指定的摩尔纹信息嵌入到一张图像中，我们基于该模块设计我们的自监督学习框架。首先我们生成大量的随机水印信息序列，将这些随机序列与基础数据集送到摩尔纹生成器中，摩尔纹生成器会根据水印信息在基础图像上生成对应的摩尔纹图像。我们再将这些生成的摩尔纹图像与控制它们生成的水印信息作为标签一起送到水印提取网络进行训练。正如前文所说，我们在训练的时候使用数据增强技术，以及在相同的基础数据集图像上嵌入不同水印信息的摩尔纹，这样能够更好地引导水印提取网络去学习到摩尔纹与水印信息的对应关系，而忽略图像本身的信息。我们的训练可以使用下面的流程图进行表示。

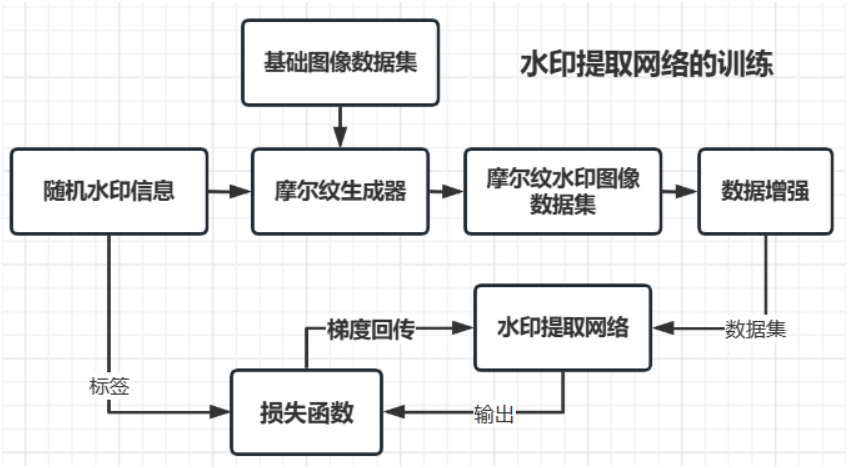

图 18: 水印提取网络的训练

网络训练时，我们不断尝试多个学习率，最终选择学习率为 1e-4 作为学习率进行训练，由于信息的提取归根到底是一个分类问题，分类问题较好的损失函数是交叉熵损失函数，我们按照比特为单位，一个图像分块的摩尔纹可能提取出 0 信息，也可能提取出 1 信息，所以水印提取网络可以看作是不断地在进行多个二分类问题，我们将随机水印序列按照比特为单位与水印网络的输出标签计算交叉熵作为损失，并计算损失函数对模型参数的梯度，将梯度回传到每一个模型参数上去更新模型参数，得到一个更优的模型。其中交叉熵损失函数的定义如下：

$$L = \frac{1}{N}\sum_{i} L_i = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic}\log(p_{ic})$$

其中：
- M ——类别的数量
- y_{ic} ——符号函数（0 或 1），如果样本 i 的真实类别等于 c 取 1，否则取 0
- p_{ic} ——观测样本 i 属于类别 c 的预测概率

图 19: 交叉熵损失

经过训练的网络在经历 200epoch 后，得到了一个在测试集上比特提取准确率达到 92% 的模型，但是经过查看模型训练的损失函数曲线发现，网络在 60 多 epoch 之后，损失函数值一直处于左右摇摆的过程，这种情况往往是因为模型优化的步长过大，导致陷入一个极值点的附近来回横跳，但是一直无法进一步靠近极值点，因此我们决定对模型原来的训练数据集上进行微调，我们设置 1e-6 作为新的学习率，让水印模型在较小的学习率上训练 200epoch，实验结果在预期之内，模型的损失函数值能够逐步下降，测试集上的比特提取准确率提升到了 97%。

为了进一步排除模型超参数对测试数据集上图像的过拟合情况，我们还准备了与测试集等大的验证集数据集并用于模型泛化能力测试，结果显示，在验证数据集上，模型的比特提取准确率依旧能够达到 97%，说明经过了训练和 finetuning 的水印提取网络具有很好的模型泛化能力。

注：以上测试结果并不经过摄屏失真的处理，因为大量图像的摄屏处理会耗费大量的人力和时间，因此我们无法在有限时间内完成此数据集处理任务。

2.5.3 基于 vote 的冗余嵌入校验方案

实际上，仅仅通过自监督学习训练得到的水印提取网络在面对摄屏失真时，依旧会有较大的比特错误率，这是因为我们的摩尔纹嵌入算法并不是生成一个与图像无关的摩尔纹，生成的摩尔纹与图像原本

像素有较强的联系，按照摩尔纹生成计算公式，如果图像某个分块本身是偏向于白色，那么生成的摩尔纹视觉上将会非常弱，（下文称为弱摩尔纹），弱摩尔纹被水印网络提取出准确信息的概率较低，因此容易导致水印信息的残缺。

我们参考相关抗摄屏水印的论文，采用了 TERA[10] 中的冗余嵌入与交叉验证的思路，我们将一个水印的 N 个副本嵌入图像，其中 N 是一个冗余系数，这样，只要 N 个水印副本中有任意一个能够正确提取出水印的比特信息，就能实现水印的正确提取。冗余嵌入的使用大大增强了水印的鲁棒性。在提取

Algorithm 2 Vote method for redundant watermark extraction

Input: The output vectors of watermark extraction network $V_1, V_2, ..., V_n$, Watermark embedding redundancy N

Output: vote result bit $x \in \{0, 1\}$

1: **for all** i $from$ 1 to N **do**
2: $Label_i = argvmax\{V_i\}$
3: **end for**
4: $voteBox_0 = 0, voteBox_1 = 0$
5: **for all** i $from$ 1 to N **do**
6: $voteBox_{Label_i} = voteBox_{Label_i} + 1$
7: **end for**
8: **return x :** $argvmax\{voteBox_0, voteBox_1\}$

时，我们首先借鉴了 KNN 中投票的方式 vote 来对提取出的水印副本信息逐个比特进行投票，取票数多的比特信息作为最终的水印信息。但是实际的测试结果发现，vote 的方案表现并不好，对冗余纠错的贡献几乎为 0。我们仔细分析了所有冗余信息，发现，在参与投票的所有水印副本中，如果对某一个比特的摩尔纹存在争议时，将会有更多的投票者会将这一票投给错误的识别结果。这种 vote 的制度将每一个冗余水印的权重设置相等，没有进一步考虑摩尔纹失真时水印提取网络的识别特点，因此，我们进一步改进了 vote 方案，设计新的 poss 冗余校验方案，取得良好的鲁棒性效果，我们将在下一小节中详细介绍 poss 方案。

2.5.4 基于输出层激活值改进的 poss 冗余校验方案

经过仔细研究数据集的情况和水印网络输出层节点激活值的情况，我们有以下发现：

1) 被误分类的摩尔纹图像都是一些弱摩尔纹，视觉上很难感知，其容易被误分类，并且在投票中占据席位优势。

2) 被误分类的摩尔纹，水印提取网络输出层节点的激活值很小，一般绝对值不超过 2，说明水印提取网络对该图像中的摩尔纹识别把握度不高。

3) 强摩尔纹能够被正确识别，并且其输出层节点的激活值要远高于被误分类的弱摩尔纹，一般绝对值超过 10，说明水印网络对强摩尔纹有比较强大的信心将它准确无误提取出来。

基于这个观察，我们的思路是，要削弱席位优势在投票机制中的作用，并增强基于置信度的权重在投票中所发挥的优势，并且当置信度与席位投票冲突时，置信度要有足够的权重去改变席位投票的结果。

综上所述，我们首先设置一个锦标赛窗口，该窗口过滤掉置信度特别低的输出节点，实际上是剥夺了高错率摩尔纹在最终投票的席位，接着，在投票时，置信度更高的席位拥有更多的投票数目，具体的投票数量由函数 f 进行决定。f 是一个与置信度正相关的函数。算法如下所示。

2.6 摄屏几何失真的矫正

2.6.1 矫正原理

倾斜矫正基于透视变换，利用透视中心、像点、目标点三点共线，按照透视旋转定律使承影面（透视面 or 原始图）绕迹线（透视轴）旋转某一角度，破坏原有的投影光线束，仍能保持承影面上需要投影的

Algorithm 3 poss:a improved vote method for redundant watermark extraction

Input: The output possibility vectors of watermark extraction network $P_1, P_2, ..., P_n$, Watermark embedding redundancy N, poss Window size S $S \leq N$

Output: poss_vote result bit $x \in \{0, 1\}$

1: **for all** i $from$ 1 to N **do**
2: $largerposs_i = max\{P_i\}$
3: **end for**
4: get S largest possibility of watermark prediction:$T_1, T_2, ..., T_s$ from largerposs,make $T_1 \leq T_2 \leq ... \leq T_s$ and make $Label_k$is the label of T_k $k \in \{1...S\}$
5: $voteBox_0 = 0, voteBox_1 = 0$
6: **for all** i $from$ 1 to S **do**
7: $voteBox_{Label_i} = voteBox_{Label_i} + f(T_i)$, function f is the confidence weight calculation function, which is generally positively correlated with the confidence level
8: **end for**
9: **return x :** $argvmax\{voteBox_0, voteBox_1\}$

集合图像不变。体现出来的效果为将一张斜视角图片变为俯视角图片。

透视变换常用于视觉导航研究中，由于摄像机（or 声呐等摄影设备）和地面（or 海底）之间有倾斜角，不是直接垂直朝下进行的正向投影，有的工程应用希望将图像矫正为正向投影，就需要利用透视变换。

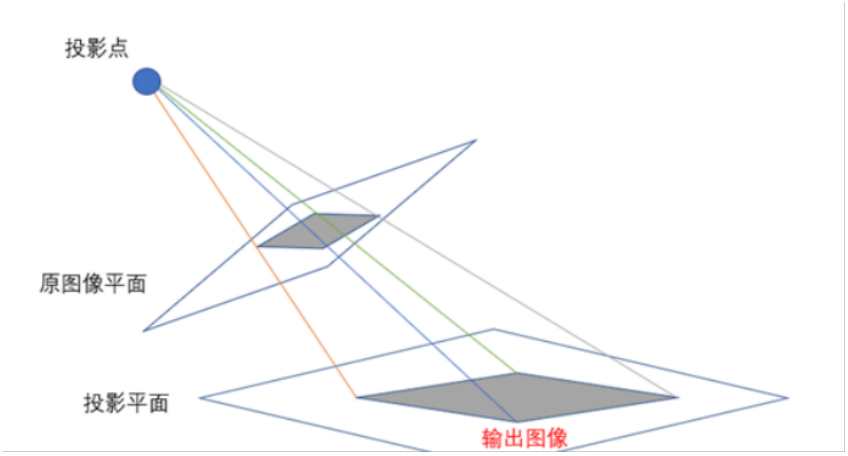

图 20: 失真原理

在我们的项目中，矫正能将待提取图片变换为垂直视角，能有效提高提取正确率。

2.6.2 透视变换的实现

透视变换是将图片投影到一个新的视平面，也称作投影映射。它是二维到三维，再到另一个二维空间的映射。相对于仿射变换，它不仅仅是线性变换。它提供了更大的灵活性，可以将一个四边形区域映射到另一个四边形区域。透视变换也是通过矩阵乘法实现的，使用的是一个 3x3 的矩阵，矩阵的前两行与仿射矩阵相同，这意味着仿射变换的所有变换透视变换也可以实现。透视变换的效果相当于观察者的视角发生改变时所观察到画面产生的变化。

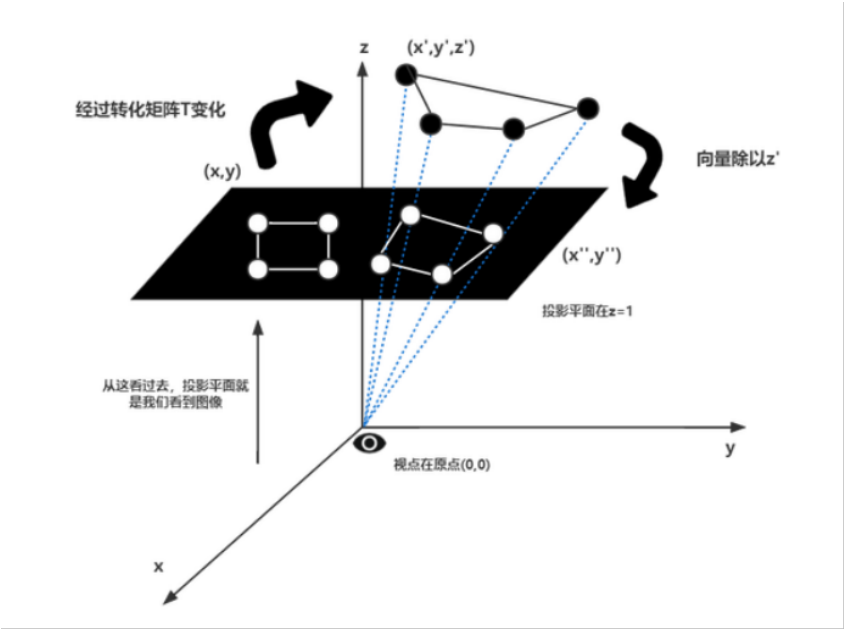

图 21: 透视变换

即透视变换引发我们图片视角的变化并不是我们观察者视角的变化，而是观察者视角不变而对物体空间位置发生了转换，导致了我们看到的图片视角变化。这可以理解为：不是我们人走动看到了物体角度发生变化，而是人始终保持不动，其他人对物体进行了挪动导致我们看到这个物体角度发生了变化。

2.6.3 关键代码解读

在实现中，需要用 OpenCV 的两个函数 getPerspectiveTransform() 和 warpPerspective() 两个函数。

getPerspectiveTransform()：此函数用于从 4 对映射点计算透视变换的变换矩阵 T，返回的矩阵数据类型是 Mat。

1) 参数 src：源图像四边形的 4 个顶点坐标。

2) 参数 dst：目标图像对应四边形的 4 个顶点坐标。

3) 参数 solveMethod：传递给 cv::solve(DecompTypes) 的计算方法，默认是 DECOMP_LU。

4) 返回值：Mat 型变换矩阵，可直接用于 warpPerspective() 函数

warpPerspective()：此函数用于将变换矩阵 T 应用于原图像，使其透视变换为目标图像。

```
void warpPerspective(
    InputArray src,
    OutputArray dst,
    InputArray M,
    Size dsize,
    int flags=INTER_LINEAR,
    int borderMode = BORDER_CONSTANT,
    const Scalar& borderValue = Scalar());
```

图 22: 关键代码

1) 参数 src：输入图像。

2) 参数 dst：输出图像，需要初始化一个空矩阵用来保存结果，不用设定矩阵尺寸。

3) 参数 M：3x3 的转换矩阵。

4) 参数 dsize：输出图像的大小。

5) 参数 flags：设置插值方法。默认为 INTER_LINEAR 表示双线性插值，INTER_NEAREST 表示最近邻插值，WARP_INVERSE_MAP 表示 M 作为反转转换 (dst->src) 。

6) 参数 borderMode：像素外推方法，默认为 BORDER_CONSTANT，指定常数填充。翻阅官方文档发现还有一个选项是 BORDER_REPLICATE。

7) 参数 borderValue：常数填充时边界的颜色设置，默认是 (0,0,0)，表示黑色。这就是为什么透视变换后图片周围是黑色的原因。这里需要注意的是类型为 Scalar (B, G, R)。

图像分析，矫正方向的一个重难点为确定透视内容，即选取被矫正区域，我们采用手动确定的方式。通过手动确定矩形区域四个顶点，自动根据图片坐标生成被矫正区域，再经过上述方法自动矫正。

```
#手动部分
cv2.setMouseCallback('origin', on_mouse)  # 此处设置显示的图片名称一定要和上一句以及on_mouse函数中设置的一样
cv2.waitKey(0)  # 四个角点点击完后，随机按键盘结束操作
cv2.destroyAllWindows()
```

图 23: 关键代码

```
# 在点击图像处绘制圆
# cv2.circle(image, center_coordinates, radius, color, thickness)
cv2.circle(img2, p1, 4, (0, 255, 0), 4)
```

图 24: 关键代码

2.7 基于 Hamming 的水印信息序列生成与校验

海明码是一种具有检错纠错能力的二进制编码，对于 1bit 错误非常有效，同时需要添加的编码数量比较少。

2.7.1 Hammig 码的生成

流程如下：

1) 确定校验码的位数：设数据有 n 位，校验码有 x 位。则校验码一共有 2^x 种取值方式。其中需要一种取值方式表示数据正确，剩下 $2^x - 1$ 种取值方式表示有一位数据出错。因为编码后的二进制串有 n+x 位，因此 x 应该满足 $2^x - 1 \geq n + x$

2) 确定校验码的位置: 校验码在二进制串中的位置为 2 的数幂

位置	1	2	3	4	5	6	7	8	9	10	11
内容	x1	x2	1	x3	0	1	0	x4	1	1	0

图 25: 校验过程

3) 求出校验位的值：采用偶校验:

位置	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011
内容	x1	**x2**	**1**	x3	0	**1**	**0**	x4	1	**1**	**0**

图 26: 校验过程

为了求出 x2, 要使所有位置的第二位是 1 的数据（即形如 **1* 的位置的数据）的异或值为 0。即 $x2\ xor\ 1\ xor\ 1\ xor\ 0\ xor\ 1\ xor\ 0 = 0$。因此 $x2 = 1$。以此类推可得每一位校验位的值。

位置	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011
内容	0	1	1	1	0	1	0	0	1	1	0

图 27: 校验过程

2.7.2 Hamming 码的校验

以偶校验为例，所有校验码所在的位是只由对应的校验码进行校验，如第 1 位（只由 0001 校验）、第 2 位（只由 0010 校验）、第 4 位（只由 0100 校验）、第 8 位（只由 1000 校验）……。也就是这些位如果发生了差错，影响的只是对应的校验码的校验结果，不会影响其它校验码的校验结果。这点很重要，如果最终发现只是一个校验组中的校验结果不符，则直接可以知道是对应校验组中的校验码在传输过程中出现了差错。

所有信息码位均被至少两个校验码进行了校验，也就是至少校验了两次。查看对应的是哪几组校验结果不符，通过交叉比对找出这几组共同对应信息码，就可以很快确定是这位信息码在传输过程中出了差错。

将每一位进行检错 (包括校验位) 后，对错误比特进行翻转，即可得到正确编码。在提取信息时，根据生成规律，校验位均在 2 的幂次位置，跳过这些位提取即可。

```
#去掉纠错码，提取信息
position = 1
msg = ""
for i in range(len(d)):
    j = len(d) - i - 1#
    if i == position - 1:
        position = position * 2
        continue
    msg = msg + d[j]
```

图 28: 关键代码

三、 作品测试与分析

3.1 实验环境

我们的实验环境主要部署在云服务器上，少量代码需要可视化，部署在本地 PC 计算机。具体的实验环境配置如下所示。

表 2: 实验硬件环境配置

GPU	RTX 3080（AutoDL 10GB *1）
CPU	15 核 Intel(R) Xeon(R) Platinum 8358P CPU @2.60GHz
Memory	80GB
PyTorch	1.11.0
Python	3.8
OS	Ubuntu20.04
Cuda	11.3
Matlab	R2022a

表 3: 水印提取网络训练超参数配置

数据集	自制数据集
数据集大小	2000 images 64*64
网络架构	改进的 Alexnet
训练方式	自监督学习
数据增强	Flip+Resize
训练 epoch	200+200Finetuning
learning rate	1e-4（Finetuning1e-6）
网络预训练	无

3.2 功能测试

功能测试部分主要验证我们的系统在水印嵌入和水印提取两个流程是否能够完成预期的功能。

3.2.1 水印嵌入

我们使用一张 elaTony 的图片进行测试。首先，我们选择原始的水印信息序列 0110，水印序列经过 Hamming 校验模块，得到要嵌入的水印序列 0110011。

图 29: Hamming 校验模块生成水印校验序列

现在我们使用摩尔纹生成模块去在原始图像上生成自适应的摩尔纹。嵌入水印前后的图像如下图所示。如图所示，嵌入后，从视觉上可以感受到摩尔纹的存在。尽管此时摩尔纹水印比较明显，但是一旦经历摄屏场景后，摩尔纹会与摄屏天然产生的摩尔纹混为一体，几乎难以分辨，我们的水印就会显得非常自然。

图 30: 嵌入前原始图像

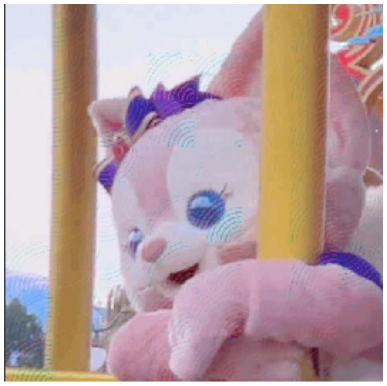

图 31: 嵌入后的水印图像

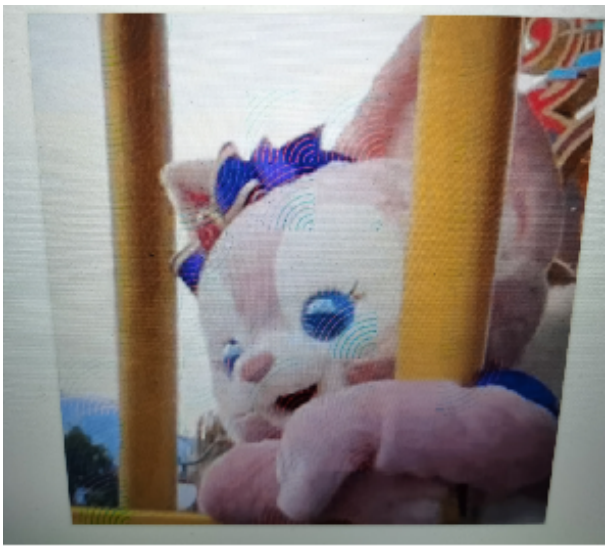

图 32: 摄屏后的图像

3.2.2 水印提取

经历摄屏后，图像发生了形变的几何失真，光照带来颜色和亮度的失真，还有摄屏场景下的摩尔纹和其它的高斯噪声。我们需要将图像先进行几何矫正，让图像恢复正面形状。下面是我们矫正前后的图像对比。现在我们将图像输入水印提取网络，水印网络已经集成了冗余校验 poss 和 Hamming 校验，我

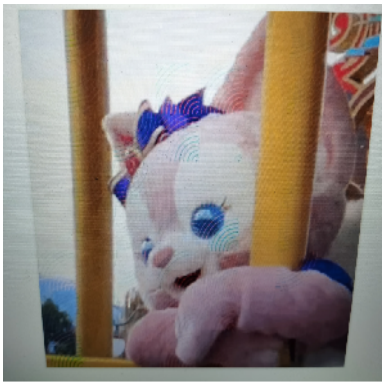

图 33: 摄屏图像

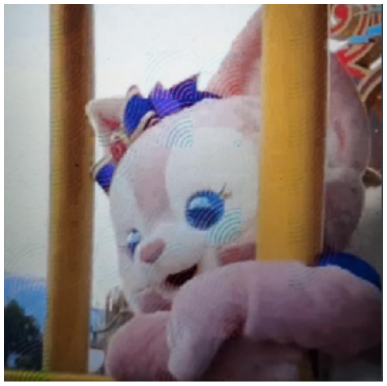

图 34: 矫正图像

们可以从程序的输出中直接获得提取出来的水印信息，如图所示，我们的水印在 poss 方案下已经能够以

100% 的准确率提取出来了。

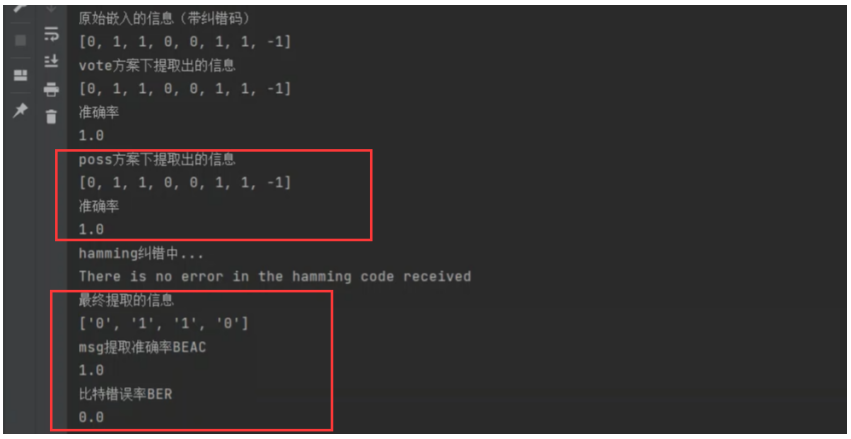

图 35: 提取出的水印

我们为上述流程录制了一个演示视频，已经随本项目报告一起提交。

3.3 性能测试

3.3.1 水印的鲁棒性

我们使用水印信息 0110 得到的校验码 0110011 对 16 张真实图像进行测试实验，包括多个角度的摄屏实验，实验结果如下所示，可以看到，我们的作品在水印鲁棒性上具有很好的性能，经过冗余校验和 Hamming 纠错码对水印信息的纠错，水印比特提取准确率几乎接近 100%。

表 4: 水印鲁棒性测试

嵌入容量	64*64*N per bit(N 是冗余参数，推荐 N=8)
vote 方案提取准确率	83.33%
vote 方案提取准确率（含纠错）	（vote 已抛弃）
poss 方案提取准确率	91.67%
poss 方案提取准确率（含纠错）	100%
倾斜拍摄 45 度以内提取准确率（含纠错）	100%
剪裁 20% 以内提取准确率（含纠错）	100%

此外，我们对图像进行了不同角度的拍摄以及不同光照条件下的拍摄，结果发现，当倾斜角度不超过 45 度时，水印的鲁棒性几乎不会受到影响；当光照对图像的影响不至于导致整个比特的所有冗余摩尔纹分块全部难以分别辨认时，水印的提取依旧可以成功，由于冗余水印的存在，在面对图像小规模裁剪时依旧可以保持一定的鲁棒性。

3.3.2 水印的隐蔽性

从视觉上看，经过摄屏后的图像在视觉上几乎很难看出明显的差别。

从统计特征上看，我们使用课堂上学习的隐写分析手段卡方分析对水印图像进行检测，检测结果如下图所示，由此可知，我们的水印图像在面对卡方分析的水印检测时具有很强的隐蔽性。

图 36: 无水印

图 37: 有水印

表 5: 卡方分析的水印检测测试

检测手段	卡方检测
p 值	0.000
是否检测出水印信息	无

3.3.3 嵌入容量

本项目中，默认规定 64*64 像素的图像摩尔纹表示一个比特的信息，并且由于冗余校验的存在，设冗余系数为 N，一个比特需要的图像像素空间为 64*64*N。假设一张图像的大小一共有 m*n 个像素，我们采用全行。全列冗余嵌入的方式，那么该图像可以嵌入的比特信息有 $min\{floor(m/64), floor(n/64)\}$。这样的嵌入容量相对来说是比较低的。

从嵌入率的角度看，由于我们使用的是 Hamming 编码，我们的嵌入率就是 Hamming 的嵌入率，Hamming 的冗余位是满足下面不等式的最小的 k 值，

$$2^k - 1 \geq n + k$$

在 k 很大的情况下，我们可以认为冗余位与信息位是一种对数函数关系，k 远小于 n，因此嵌入率相比于其它校验码还是很高的。

3.3.4 时间性能

我们对水印嵌入与提取两个流程的主要环节进行了时间性能的测试，相关数据如下表所示，可以看出，我们的水印系统从水印生成到水印提取消耗的时间几乎可以忽略不计，这为我们的作品在小型算力系统中的部署和大规模水印图像生成应用提供了可行性。

表 6: 时间性能测试

子流程	水印信息生成摩尔纹并嵌入图像	生成校验码	图像几何失真矫正	水印提取网络提取水印信息	水印信息校验
系统部署	PC 端	服务器端	PC 端	服务器端	服务器端
运行时间	0.0011s/bit	0.0003s/4bit	0.780s	0.3s/bit	0.0001s/4bit

四、 创新性说明

本作品具有以下创新性。

1. 水印隐蔽性：本作品突破隐蔽性等价于视觉一致性的传统思维惯性，发掘出水印的自然性也能达到欺骗视觉感知，从而达到水印隐蔽这一特点。因此我们方案的隐蔽性不来源与水印信息的不可见，而是水印信息的不可察觉。本作品立足于摄屏场景，将水印信息伪装成摄屏过程中自然产生的摩尔纹嵌入图像，在不影响机器识别的基础上，能有效提高人眼视觉的隐蔽性。考虑到摄屏的使用场景大部分为人工手动拍摄，能有效避免被人眼察觉更加重要。

2. 水印鲁棒性：本作品为了实现高鲁棒性的水印提取，采用数据增强的方式获得大规模的数据集，并基于自监督学习的方式训练强大的水印提取网络进行水印提取。为了进一步弥补水印提取网络面对摄屏这一特殊场景中的透镜畸变，光畸变，几何畸变，摩尔纹等严重失真带来的提取准确率不足，我们增加了几何失真矫正、改进的 poss 冗余校验方案和 Hamming 校验码纠错等功能模块，实现水印提取极强的鲁棒性。值得一提的是，我们通过观察神经网络输出神经元激活值的特点，在传统 vote 的方案上加以改进，设计了基于置信度的 poss 冗余校验方案，有效解决弱摩尔纹在校验 vote 阶段决策席位过多的问题。

五、 总结、分工与开源情况说明

5.1 总结

针对目前摄屏技术的广泛应用带来的版权保护、泄密追溯和秘密通信的问题，我们提出了一种全新的抗摄屏水印方案，从信息的组织、载体的选择、嵌入提取策略着手，我们利用所学知识和数据库资源，探索有效的解决方法，保证水印具有足够的隐蔽性和嵌入量，大幅提高鲁棒性。我们使用摩尔纹水印作为信息嵌入图像，不论是嵌入信息后的图片还是拍摄后的含水印图片都具有一定的隐蔽性；同时将图像分成多分 64*64 的小块用于存储 1 位信息，提高鲁棒性的同时兼顾嵌入率。我们的水印逐比特进行嵌入和提取，利于提高分类器的训练、判断效率与准确性。为了进一步提高模型的提取准确度，我们在构建模型时，使用 poss 方案取代 Alexnet 网络中传统的 vote 方案；在训练模型时，利用数据增强技术扩充数据集，例如将一份训练/测试样本进行左右上下反转和旋转得到多份样本，对模型进行训练；在进行提取时，利用几何形状矫正技术定位拍摄的图像主体，使用图像主体作为水印提取网络的输入，降低外界干扰。而且，我们还使用 hamming 码对嵌入信息进行特殊编码，进一步提高信息的容错与纠错能力。我们的方案具有以下三大特点：

水印自然: 本项目立足摄屏场景，抓住摄屏过程的自然产物摩尔纹作为切入点，将水印伪装成摩尔纹嵌入图像，水印显得自然隐蔽。

水印鲁棒: 本项目引入自监督学习训练的深度神经网络作为水印提取器，辅以几何矫正、冗余嵌入和 Hamming 纠错码技术，有效抵御摄屏中的各种失真，鲁棒性强。

系统弱耦合：本项目代码模块划分清晰，具有很强的扩展性，支持自主重训练网络、数据集替换，自主设计水印，选择校验方式等拓展功能。

5.2 未来工作

由于我们采用图像分块的方式嵌入信息，这样嵌入后的图像存在水印割裂的特点，不利于隐蔽性。在项目的最后，我们讨论的可行解决方案是不对图像进行分块，而是用整个图像作为嵌入单位信息的载体。同时，增加水印的类型（即增加嵌入算法中可使用参数的数量）。例如，我们图像的 4 个角及 4 条变的重点作为将可选择的原点，那么一张图像就可以嵌入信息 0-7, 此时的信息单位位 3bit。这是一种牺牲嵌入量来提供隐蔽性的改进措施，而且，对分类器也有一定的挑战。我们将在未来的工作中尝试这种方式。

5.3 小组分工情况

马锦贵：本次项目作为组长，负责项目选题，团队协调，把握进度，是项目主要创新点摩尔纹伪装水印的提出者。代码实现主要负责水印提取网络结构设计，自监督学习训练，数据增强技术优化，改进冗余校验方案等工作，是关键优化点 poss 方案的发现和实现者，冗余校验的引入使得水印提取准确率从 77% 上升到 87.5%，poss 方案使得水印提取准确率从 87.5% 上升到接近 100%。

熊银川：本次项目组员，负责对嵌入信息进行纠错编码和对图片预处理。代码实现部分为设计实现 Hamming 码的生成，校验，检错，纠错，提高信息提取正确率。设计实现图片的透视矫正，用于应对真实摄屏场景，以及用作不同倾斜角度的提取效果比对，实现对非俯视图片的信息提取。

王仕信：本次项目组员，前期负责摩尔纹调研工作，介绍了摩尔纹的定义、价值以及原理。后期负责设计一个自适应的图像摩尔纹效果算法，并使用该算法实现水印信息在图像上的隐写，为项目的实现准备了多个水印嵌入方案并提供对应的大量自制数据集。

周正业：本次项目组员，负责实验结果的测试以及相关参数的调整。

5.4 项目开源情况说明

本项目的所有源代码，数据集，环境配置等将于 2023 年 5 月-6 月进行开源，需要学习参考开源提前联系我 2358231418@qq.com 开源地址：https://github.com/majingui/shooting-watermark-IHproject.git

本项目中水印提取网络从设计，实现到训练，优化，vote 与 poss 冗余校验方案的每一行代码均由本小组自己完成，没有使用任何开源项目代码。摩尔纹生成与嵌入模块，Hamming 校验，几何矫正模块借鉴了开源项目的代码并进行改进优化。

本项目借鉴的其它开源项目代码情况如下所示，特此感谢。

摩尔纹 [27] https://blog.csdn.net/matrix_space/article/details/42215233

Hamming 纠错 Hamming code[12] https://github.com/absingh31/HammingCode

几何矫正 https://blog.csdn.net/weixin_42410915/article/details/120280933

参考文献

[1] Han Fang, Zhaoyang Jia, PIMoG: An effective screen-shooting noiselayer simulation for deep-learning-based watermarking network. In MM ' 22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022,pages 2267–2275. ACM, 2022.

[2] Fang H, Chen D, Wang F, et al. TERA: Screen-to-Camera Image Code with Transparency, Efficiency, Robustness and Adaptability[J]. IEEE Transactions on Multimedia, 2021, 24: 955-967

[3] Cui H, Bian H, Zhang W, et al. Unseencode: Invisible on-screen barcode with image-based extraction[C]//IEEE INFOCOM 2019-IEEE Conference on Computer Communications.

[4] Fang H, Zhang W, Ma Z, et al. A camera shooting resilient watermarking scheme for underpainting documents[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019.

[5] Fang H, Zhang W, Zhou H, et al. Screen-shooting resilient watermarking[J]. IEEE Transactions on Information Forensics and Security, 2018, 14(6): 1403-1418.

[6] Alexnet:ImageNet Classification with Deep Convolutional Neural Networks

[7] Self-supervised Visual Feature Learning Deep Neural Networks: A Survey , Longlong Jing, Yingli Tian

[8] Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation Xiangyu Yue,Zangwei Zheng,Shanghang Zhang,Yang Gao

[9] PIMOG https://github.com/FangHanNUS/PIMoG-An-Effective-Screen-shooting-Noise-Layer-Simulation-for-Deep-Learning-Based-Watermarking-Netw

[10] TERA details https://www.researchgate.net/publication/349601158_TERA_Screen-to-Camera_Image_Code_with_Transparency_Efficiency_Robustness_and_Adaptability

[11] Image math correction code https://blog.csdn.net/weixin_42410915/article/details/120280933

[12] Hamming code https://github.com/absingh31/HammingCode

[13] Hamming correction details https://blog.csdn.net/huoji555/article/details/103244830

[14] Deep learning model Alexnet details https://blog.csdn.net/hongbin_xu/article/details/80271291

[15] Moire generation details https://blog.csdn.net/matrix_space/article/details/42215233

[16] Data Augmentation https://zhuanlan.zhihu.com/p/41679153

[17] Self-Supervised Learning https://zhuanlan.zhihu.com/p/184995155

[18] Self-Supervised Learning details https://blog.csdn.net/sdu_hao/article/details/104515917

[19] Pytorch and DeepLearning https://www.bilibili.com/video/BV1hE411t7RN?vd_source=543b1fdd118d8898b4c529bef1395239

[20] Moire https://zhuanlan.zhihu.com/p/21478979

[21] CNN https://blog.csdn.net/qq_25762497/article/details/51052861

[22] CNN https://www.jianshu.com/p/70b6f5653ac6

[23] 刘芳龙. 基于无监督学习的屏摄图像去摩尔纹研究 [D]. 天津: 天津大学,2020.

[24] 李桐. 感知一致的图像摩尔纹去除方法 [D]. 山东: 山东大学,2020.

[25] 栗小斌. LED 显示屏摩尔纹的消除方法 [J]. 演艺科技,2013(8):44-47. DOI:10.3969/j.issn.1674-8239.2013.08.011.

[26] 钱恺. 以 LED 大屏为背景时摩尔纹的产生机制分析 [J]. 有线电视技术,2019,26(5):38-40.

[27] 摩尔纹 https://blog.csdn.net/matrix_space/article/details/42215233

教师评语评分

评语：

评　　分：

评 阅 人：

评阅时间：