

# 版权护卫

伴随着互联网和图像技术的快速发展，人们的生活与工作变得越来越方便，交流和娱乐也越来越多样化，但这其中同时也带来了不少安全隐患。回想过去十年的中国互联网暴力野蛮的发展，数字图像的易拷贝和易传播特性，带来了大量的侵权和盗版问题，给著作人的利益带来了严重的损害，对大量的数字行业带来的深刻并且严重的破坏。幸运的是，现在版权问题已经获得越来越多的人重视，社会和国家也在促进版权保护的发展，各个公司更是视版权为财富。即便如此，由于传统的可见水印嵌入方案已显乏力，不能跟上时代对版权的号召，存在着大量的安全漏洞，可见水印对版权的保护形同虚设。本平台将鲁棒的不可见水印技术和互联网技术进行结合，为此类侵权盗版问题的解决提供了新的视角。虽然不可见水印技术目前仍然存在不足，没有大量的商业化，也不能彻底解决存在的安全问题，但是其提供了一个较为新颖的解决方案，并可以提供比可见水印更强的版权保护。

## 一、小程序说明

### 1.1 产品简介

本文产品的名称叫做“版权护卫”，以鲁棒的不可见水印技术为核心，构建了一套精简的版权保护平台，为用户提供最基本的版权保护能力的同时，也能保障图像的观赏性，甚至在图片在一定程度破坏的情况下，同样能够提取出水印。图 1 是本产品的首页。

【图 1】

### 1.2 背景和现状

随着数字技术和通信技术的发展，图像成为目前互联网中最受欢迎的多媒体形式之一。2010 年以后，智能手机的大面积普及，也让越来越多的人通过手机拍照来记录生活点滴，再加上手机中各类滤镜的使用与后期手段，手机已经可以拍出不逊色于专业相机的效果，因此人们热衷于通过移动端的操作来进行图片的制作和分享。

发达的数字图像技术和便捷的分享途径，在给人们带来丰富乐趣和充实的图像内容的同时，也隐藏着多媒体安全问题。在互联网平台上发布自己创作的图片时，容易遭它人盗用，这将会极大侵害图片创作者的个人权益。现阶段，作者在发布图像作品时，为了达到最基本的版权保护功能，通常会往图片中加入可见的水印，以起到保护的作用，水印中通常包含作者或图像本身的相关信息，如图 2 中的照片加入了可见的水印信息“大阪”。



图 12 普通水印

可见水印的引入,改善了图片侵权的现状,也在一定程度上促进了大众的版权保护意识,但是可见水印也存在诸多的缺点亟待解决:

1).安全性过低。数字图像修复技术可以对破损图片进行一定程度的修复,水印本身就是一种可见的图片破损,因此水印极易被包括数字图像修复技术在内的相关技术进行去除,使得不法分子还原出高质量的原图,进而满足自己的利益。对于一些放置于边边角角的水印而言,直接可以通过截图技术就能去掉水印。

2).对视觉的影响。为了保证安全性,会采用将图像放置图片中央,甚至布满图片的形式,如图 3 所示。这样子的可见水印虽然提高了安全性,不易被截去,但是也对视觉效果带来了较大的破坏,影响了图片的观赏性,不利于图像放在互联网平台上的初衷“传播”。

3).真实性问题。为了提升安全性并且降低对视觉的影响,现在越来越多图像创作者喜欢在图像中引入花式的水印,这些水印看起来较为真实,不容易被攻击者理解为水印,进而被去除,如图 4 所示,碗上的“大阪”标识实际上为水印。虽然这样可以确实能起到效果,但是若被理解为水印,同意也容易被去除,并且容易让观赏者误会。

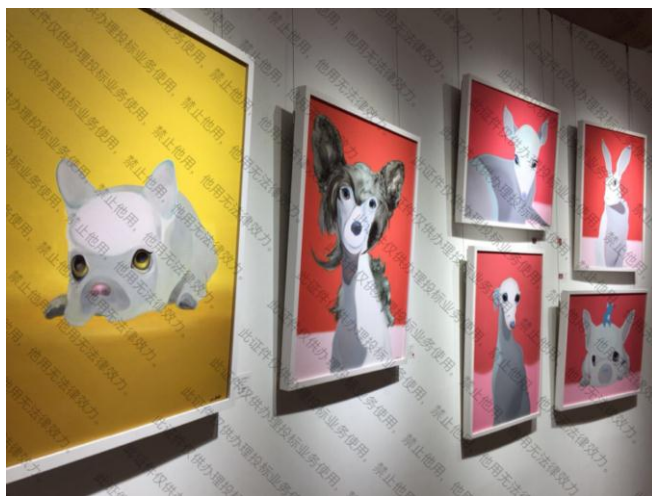


图 3 强可见水印



图 4 花式可见水印

为了解决可见水印存在的诸多问题，学术界在早期就已经提出了包含 LSB 在内的不可见水印技术，但是由于技术不够成熟，此类技术持续沉浸于实验室多年。随着近几年鲁棒不可见水印技术突飞猛进的发展，不可见水印越来越丰满和成熟，已经具备一定的商业使用价值。图 5 中分别显示了原始图像、水印图像和含水印的图像，很明显，从含水印图像中完全无法看出关于水印的信息。



图 5 原始图像、水印图像与含水印图像

### 1.3 支持的功能

#### 1.3.1 不可见水印

该功能可以往图像中嵌入不可肉眼察觉的水印，需要在选择了原始图片后，输入图片的标题(选填)，不可见水印文字内容(必填)，提取水印的密钥(必填)。由于不可见水印涉及到较为复杂的计算，通过手机来进行运算速度太慢因此当前版本会将图像发送到服务器进行不可见水印内容的嵌入，嵌入完成后会将含密水印返回。图 6 描述了不可见水印嵌入的相关页面。

(a).输入必要信息 (b).等待服务器嵌入完成 (c).服务器返回的图像

【图 6 嵌入不可见水印】

存在水印的嵌入操作，就会有水印的提取操作，需要在选择了图像后，输入水印密钥。图 7 描述了不可见水印提取的相关页面。

【图 7 提取不可见水印】

#### 1.3.2 图片注册和追溯

图片注册和追溯，是一个闭环的版权保护功能，当图片创作者在本平台上进行了照片的注册后，平台将会绑定图片和作者，并返回一个注册图片(该图片含水印)，作者可以将服务器返回的注册图片用于任何活动的使用。在图 8 给出了图片注册的相关页面。

【图 8 图片注册】

在平台上，用户对注册图片进行溯源，能够找到该图片所对应的注册者，并和该注册者取得联系。在图 9 中给出了图片追溯的相关页面。由于微信是一个较为私人敏感的平台，因

此为了避免注册者的隐私遭到泄露，因此追溯成功是不会显示关于注册者任何相关信息的。

【图 9 图片追溯】

### 1.3.3 留言收发

对于在平台上注册的图片，用户能够在平台上进行图片追溯，以找到图片的注册者，当找到注册者以后，用户可以向注册者发送消息。为了避免消息发送者的隐私泄露，本平台不会发送关于发送者的任何隐私，因此需要用户在消息中写入自己的愿意透露的别名和联系方式等信息。图 10 中给出了发送留言的相关页面。

【图 10 发送留言】

在留言发送成功后，注册者会接收到未读的留言，注册根据留言发送者中透露的信息，自行选择是否和该用户进行联系。在图 11 中显示了留言查阅的相关页面。当前版本还未引入黑名单功能，在后续会加入。

【图 11 留言查阅】

### 1.3.4 可见水印

本产品虽然是以不可见水印技术核心的，但是为了更为平稳的往不可见水印技术进行过度，并且提供更广全方面的版权保护功能，因此同样也提供了可见水印的功能。可见水印的功能较为简单，并不会涉及到复杂的数学计算，因此在小程序端进行完成，并支持离线功能。在图 12 中给出了可见水印的相关页面。

【图 12 可见水印】

### 1.3.5 历史记录

考虑到用户进行了水印嵌入操作以后，由于某些原因导致了含水印图像的丢失，因此在服务器上保存了含水印的图像，用户可以通过历史记录查看所有操作的含水印图像，需要注意的是服务器中没有保存用户提供的原图，历史记录中也不会提供原图的记录，并且也不会包含可见水印图像的记录。在图 13 中给出了历史记录的相关页面。

【图 13 历史记录】

## 二、应用场景

## 三、解决的问题

### 3.1 版权保护

### 3.2 盗版追踪

## 四、技术开发方案

### 4.1 技术选型

#### 4.1.1 小程序端

#### 4.1.2 服务器端

1).Python

2).C++

3).Flask

4).MySQL

5).Linux

#### 4.1.3 核心算法

1).Matlab

2).OpenCV

## 4.2 解决方案

### 4.2.1 小程序端

### 4.2.2 服务器端

- 1). 会话机制
- 2). 异常机制
- 3). 限流机制
- 4). 图片注册和追溯机制
- 5). 邀请发送机制
- 6). 变分辨率问题

### 4.2.3 核心算法

## 五、团队的组成与分工

### 5.1 陈智隆(队长)

负责鲁棒水印核心算法编写，包括了：1).测试微信发送图片对图片本身的 JPEG 压缩和变分辨率情况。2).查阅相关文献，进行鲁棒水印的 Matlab 仿真，并测试在微信发送后水印的提取情况。3).采用 OpenCV 库，将鲁棒水印算法从 Matlab 移植到 C++平台，并分别编译了 Windows 和 Linux 版本的动态链接库，以便 Python 调用。

### 5.2 李素静

负责小程序端的界面显示以及所有交互效果，包括了：1).了解并熟悉小程序端开发组件和工具。2).设计并开发了小程序的显示界面。3).完成人机交互效果，优化用户体验。

### 5.3 卢帅吉

负责系统架构的设计以及系统的开发与部署，包括了：1).通过腾讯云架设服务器运行环境，如域名备案，Nginx 搭建，Python-Flask 环境搭建，SSL 注册等。2).设计服务器的 HTTPS 接口。3).开发服务器 CRUD 业务代码。