

解决 HEIF 图片转 JPG 偏色问题

以 小米澎湃OS 相册的 `HEIF` 图片为例

1. HEIC/HEIF 简介（10秒说完）

- 一种图片格式，比 JPG 更高效，功能更多
- HEIC 是 HEIF 的一种实现，一般存照片
- HEIF 更通用，支持动态照片等

2. 为什么要转换？

- 兼容性：一些设备/看图软件不支持

比如在 Windows 11 上，除了自带的“照片”外，我试了几款软件均无法正常浏览经过小米相册编辑的 HEIF 文件（灰蒙蒙的）

3. 问题

- 已经有很多软件，脚本可以转换了，但和看图软件一样，**转换后灰蒙蒙的**

具体表现：HEIC 转 JPG 正常，HEIF 转 JPG 异常

见 7. 研究历程

4. 我的脚本能做到

- `HEIC` 转 `JPG` , `HEIF` 转 `JPG` , 正确保留色彩
- 可选转换颜色空间, 可选保留 `icc` 校色信息
- **注意:**
 - i. 编译版不保留 `exif` 信息
 - ii. `.py` 脚本版本默认保留 `exif` 信息 (`exif` 信息过长会保存失败, 此时需要将 `is_with_exif` 设为 `False`)

`exif` 信息包含保留了地理位置信息, 徕卡风格, 镜头参数等; 保留这部分信息就可以再添加徕卡水印, 或记录拍摄地点。
- **为什么不转为 `png` ? 虽然 `png` “无损”, 但 `png` 不支持 `EXIF` 信息。为了保留画质, 我的脚本保存时均设置 `quality=100`**

5. 下载 & 使用

- GitHub开源: <https://github.com/CNDY1390/HEIF-to-JPG>
 - 包含 `.py` 脚本和编译版 `.exe`
 - 如果有 Python 环境就尽量用脚本版, 自由度更高

License: MIT

后续会在评论区给出更多下载方式

5. 获取

- GitHub开源: <https://github.com/CNDY1390/HEIF-to-JPG>

- 包含 `.py` 脚本和编译版 `.exe`
- 如果有 Python 环境就尽量用脚本版, 自由度更高

License: MIT

后续会在评论区给出更多下载方式

- **希望得到您的点赞, 三连**    
- 如果你有其他问题或发现更多优化方法, 欢迎在评论区留言讨论!

6. 使用

Python脚本

1. 将脚本和要转换的图片放在同一目录。
2. 编辑 `.py` 脚本文件，填写要转换的图片的名称。
3. 运行脚本。

编译版exe

1. 运行程序。
2. 按要求操作。

注意

编译版exe 可能会误报病毒，请允许运行。如不放心，也可断网运行，或者下载源码检查。

- 下面更深入地讲解研究细节

7. 研究历程

问题发现

- 小米手机可以拍摄 `heic` , `jpg` 两种格式的照片, `heic` 格式在小米相册编辑并保存后会保存为 `heif` 文件。然而 `heif` 在小米相册和 `Windows 11` “照片”显示正常, 但是在比较简易的软件, 如 `ImageGlass` 就**显得灰蒙蒙的**。如果有徕卡水印, 底下的横幅白色明显显示为灰色。
- 小米相册在发送时可以保留 `exif` , 同时转为 `jpg` 。但终究要依赖小米相册这个软件, 并且文件名也会变, 不方便。****如何不依赖小米相册? ****网上有很多推荐的软件和方法。

不可行的方案

1. B站推荐网站: https://imagestool.com/zh_CN/convert-images

实测 `heic/heif` **均转换失败**

2. 使用Python借助pillow-heif库把HEIC格式图片转化为JPEG格式图片

本质上是借助pillow_heif库打开文件，再储存。但实际上

i. **无小米调色信息 颜色明显变得灰暗（且永久丢失）**

ii. 画质损失

3. <https://www.bilibili.com/video/BV1EhDhYDEyt>

这个视频说直接改后缀，**更是不正确的**

- 实际上，通过网上常用的办法，`Python` 调库可以将 `HEIC` 轻易转为 `JPG`
然而对于 `HEIF`，保存后的 `JPG` 颜色并不正常

研究目标

- 解决从 `HEIF` 转换为 `JPG` 时颜色偏灰的问题，并完整保留 `EXIF` 信息

开始研究

- 几个猜测

1. HEIF 和 JPG 存储色彩方式不同，调下对比度，亮度，饱和度就好了
2. HEIF 文件的 Gamma 曲线（EOTF）信息丢失或更改
3. 色域不对，一个是P3，一个是sRGB
4. EXIF 信息中包含颜色相关数据，不可丢掉
5. 小米搞错了。编辑后丢掉了某个颜色信息
6. 小米挖坑，HEIF 封装方式可能非标准，导致部分工具无法正确解析

- 按照顺序尝试

开始研究

- 几个猜测

1. HEIF 和 JPG 存储色彩方式不同，调下对比度，亮度，饱和度就好了

不对。怎么调又有一些颜色无法吻合

2. HEIF 文件的 Gamma 曲线 (EOTF) 信息丢失或更改

不对。要么太刺眼，要么更灰了

3. 色域不对，一个是P3，一个是sRGB

有进展。得知了有个东西叫做icc，校色文件

我在小米相册的apk中找到了这个文件（已上传到仓库 `backup/mi_dci_p3.icc`）

但其实还有一个办法获取，那就是随手拍一张照片，然后提取其中的icc（亲测两个文件相同），最新的脚本就是这么做的，不再依赖 `mi_dci_p3.icc` 这个文件

- **按照顺序尝试**

4. EXIF 信息中包含颜色相关数据，不可丢掉

不对。因为均不保留exif的情况下，问题仍然存在（部分软件显示正常，其他软件则不是）

5. 小米搞错了。编辑后丢掉了某个颜色信息

至今无法验证或否定，**请观众大佬们指导**

6. 小米挖坑，HEIF 封装方式可能非标准，导致部分工具无法正确解析

似乎是的，因为heif里如果是单张照片可以理解为一个只有一帧的HEVC编码的视频。但用ffmpeg无法正常解析所有数据。不确定，**请观众大佬们指导**

继续尝试

- 读取 HEIF 、 Python 转换的 JPG 、小米转换的 JPG
提取并对比： EXIF ， ICC ， 像素数据
 - 实测发现，其实 Python 转换的 JPG 就是完整保留颜色信息的，反而是小米转换的 JPG 颜色大不相同
 - 但小米转换的 JPG 反而是对的
- 问题出在哪？

- **真正的进展：Limited Range YUV**

- 随便拍一张，小米相册里拉低曝光和饱和度，制造出纯黑
然后在失败的软件上打开，发现是 $(16, 16, 16)$ ，而不是 $(0, 0, 0)$
同理发现纯白是 $(235, 235, 235)$
由此发现，涉及 色域限制 (Limited Range YUV) 问题：亮度分量 (Y) 被限制为 $[16, 235]$

- 既然 0 对应 16，255 对应 235
不难构造一个映射 $\phi: r_{\text{limited}} \mapsto r_{\text{full}}$

先考虑线性映射：
$$r_{\text{full}} = \frac{255}{235-16}(r_{\text{limited}} - 16)$$

- 发现色彩成功还原，但是 icc 还没用上

- 第二个进展：icc 校色文件的作用

- 最初 heif 的色彩信息是

sRGB/sRGB EOTF with DCI-P3 Color Gamut

也就是 P3色域 但小米转换后的 jpg 是 sRGB 的（而 PS 转换后还是 P3）

- 第二个进展：icc 校色文件的作用

- 进一步分析得到如下表格

操作	“照片”打开	读 ICC 打开	不读 ICC 打开
PS 保存副本 有 ICC	与 EXIF 基本一致	与 EXIF 基本一致	黑白正常，红色不够鲜亮
PS 保存副本 无 ICC	黑白正常，红色不够鲜亮	黑白正常，红色不够鲜亮	黑白正常，红色不够鲜亮
色彩不转换 有 ICC	与 EXIF 基本一致	与 EXIF 基本一致	黑白正常，红色不够鲜亮
色彩不转换 无 ICC	黑白正常，红色不够鲜亮	黑白正常，红色不够鲜亮	黑白正常，红色不够鲜亮
色彩转换 有 ICC	过饱和	过饱和	与 EXIF 基本一致
色彩转换 无 ICC	与 EXIF 基本一致	与 EXIF 基本一致	与 EXIF 基本一致

易得，PS 的逻辑就是不进行空间转换：原来是 P3 色域，就保留 P3 色域

综上不难看出，决定一个图片颜色的至少有以下 2 点：

1. 色彩信息本身（对于 `JPG` 格式是 `RGB`）
2. `ICC` 信息

虽然显示器颜色也决定着你最终看到的颜色，但是对于非 `HDR` 内容，差异不会大到整个图片都明显不一样，这里就暂不讨论

合格的看图软件至少会做到 1，但并不是所有软件都会做到 2
这可能也是一些照片发出去后不好看的原因之一 [未验证](#)

至此，我们终于完成了第一个任务：**保留色彩** **HEIF** **转** **JPG**。

算法:

- 如果是 **HEIC**，那么直接按照网上的常见做法，调库打开 **HEIC**，然后直接保存为 **JPG** 即可
- 如果是 **HEIF**，那么首先扩大 **YUV** 颜色范围：
 - 如果要保留 **ICC**，则仍然是 **P3** 色域，保留 **ICC** 保存为 **JPG**
 - 如果不保留 **ICC**，则先转换为 **sRGB** 色域，然后保存为 **JPG**
- 也就是说，该脚本支持小米和PS这两种不同的方案
 - 小米：转换为 **sRGB** 色域，不保留 **ICC**
 - PS：不转换色域，保留 **ICC**

代码详见仓库

然后是第二个任务：保留 EXIF 信息

- [photoshop图片导出如何保留exif信息？丢失exif信息的作品如何补上 - 知乎](#)
 - 文章指出：文件-导出-存储为web所用格式（旧版），元数据选全部，即可保留信息。**Photoshop 2025 实测不行。**
 - | 猜测：原因是PS根本无法读取 heif 的 exif 信息 未验证
- 对于 Python 方法。其实在 save 时，直接加一个存 EXIF 即可
但小米拍摄的一些照片似乎会恰好遇到 EXIF 信息太多，超出保存上限的情况
目前遇到的次数较少，故没有深入研究
 - 具体做法：在子进程里找已经编译好的 exiftool.exe 帮忙

如果大家有更好的方法敬请提出，后续可以再出一期视频讲解 🖥️ 😊

8. 遗留问题

1. 如何简单地保存超大 EXIF 信息？（见上页）
2. HyperOS 的“相册”是如何转换图片的？

9. 结语

这是我第一次分享原创脚本，代码质量不高，功能较少；

另外对图片格式也并不了解，只是将近期的一些摸索整合了一下。

同时发现，B站上相关的视频较少，**所以非常需要各位观众提出建议！**