叙述

书接上回：代码生成网站专属二维码 有宝宝反应，在有遮挡的情况下（怎么会遮挡呢？这种情况怎么能出现？我实在没考虑这种情况，但是为了更好的用户体验，即使这是一张小小的二维码图片也要根据反馈使它更完美）二维码扫不出来，好了，已解决。全部的代码可以前往 我的colab 查看。 为了保证二维码在不完全显示的情况下能够被成功扫描，不说了，直接上代码

下面是针对这些调整的改进代码：

代码

```
import qrcode
from PIL import Image, ImageEnhance, ImageSequence

# 生成二维码并提高容错级别
data = "https://www.example.com"  # 二维码内容
qr = qrcode.QRCode(
    version=2,  # 适中的二维码大小
    error_correction=qrcode.constants.ERROR_CORRECT_H,  # 高容错率
    box_size=10,  # 每个二维码格子的像素大小
    border=1,  # 边框减少
)
qr.add_data(data)
qr.make(fit=True)

# 生成二维码图片，并转换为 RGBA 模式，支持透明背景
qr_img = qr.make_image(fill='black', back_color='white').convert('RGBA')

# 将二维码的白色部分替换为透明
datas = qr_img.getdata()
new_data = []
for item in datas:
    # 将白色 (255, 255, 255) 替换为透明
    if item[:3] == (255, 255, 255):  # RGB是白色时
        new_data.append((255, 255, 255, 0))  # 透明
    else:
        new_data.append(item)

qr_img.putdata(new_data)

# 打开动态GIF图片
gif_path = "/content/your_dynamic_background.gif"  # 替换为你的GIF路径
gif = Image.open(gif_path)

# 设置需要的背景图和二维码的尺寸
new_size = (200, 200)  # 目标大小，可以根据需要调整

# 调整二维码的大小
qr_img = qr_img.resize(new_size, Image.LANCZOS)

# 用于存储处理后的帧
frames = []

# 遍历GIF的每一帧
for frame in ImageSequence.Iterator(gif):
    # 将每一帧转换为 RGBA 模式
    frame = frame.convert('RGBA')

    # 调节背景图片的透明度，保证背景不过度干扰二维码
```

Kaltoven Chen

```
    alpha = 0.3  # 设置透明度，值越小二维码越清晰，背景越不显眼
    enhancer = ImageEnhance.Brightness(frame)
    frame = enhancer.enhance(alpha)

    # 调整每一帧的大小与二维码一致
    frame = frame.resize(new_size, Image.LANCZOS)

    # 创建一个半透明的白色蒙版，进一步减少背景干扰
    mask = Image.new('RGBA', qr_img.size, (255, 255, 255, 180))  # 半透明白色

    # 将蒙版叠加到帧上
    frame.paste(mask, (0, 0), mask)

    # 将二维码叠加到帧上
    frame.paste(qr_img, (0, 0), qr_img)

    # 添加处理后的帧到frames列表中
    frames.append(frame)

# 保存结果为新的GIF，保留原GIF的帧速率和循环设置
frames[0].save(
    "qrcode_with_optimized_dynamic_background.gif",
    save_all=True,
    append_images=frames[1:],  # 添加其他帧
    duration=gif.info['duration'],  # 使用原始的帧持续时间
    loop=gif.info.get('loop', 0)  # 保留循环设置
)
```

改进点

1. **高容错率**：通过设置 `error_correction=qrcode.constants.ERROR_CORRECT_H`，二维码有 30% 的损坏空间仍然可以被正确识别。这允许部分二维码被遮盖或干扰而不影响扫描效果。
2. **背景透明度调节**：通过 `alpha = 0.3` 减少背景对二维码的干扰，保持二维码在背景上更加清晰。可以根据需要调整这个值。
3. **半透明蒙版**：添加一个半透明白色蒙版（透明度值 180），进一步减少背景图对二维码扫描的干扰，使得二维码部分更加显眼。
4. **二维码大小调整**：二维码和背景都调整为同等大小（不要过小），确保二维码有足够的像素以提高可读性。

Kaitoven Chen