تمرین تشریحی دوم چند رسانه ای

سروین نامی ۹۹۳۱۱۰۳

افزایش یا کاهش اندازه ماتریس دیترینگ در الگوریتم ordered dithering تاثیرات متفاوتی دارد. اندازه ماتریس معمولاً به عنوان یک فاکتور مهم برای کیفیت نهایی تصویر مورد استفاده قرار میگیرد. با افزایش اندازه ماتریس، تصاویر میتوانند جزئیات بیشتری را نگه دارند، اما همچنین ممکن است این افزایش باعث افزایش نویز و پیچیدگی محاسبات شود. از طرف دیگر، کاهش اندازه ماتریس میتواند منجر به از دست رفتن جزئیات تصویر شود، اما باعث کاهش نویز و سادگی محاسبات میشود.

بله، سایز ماتریس میتواند ۳ * ۳ باشد. با این حال، اندازه ماتریس کوچکتر از ۴ * ۴ ممکن است به کاهش کیفیت تصویر منجر شود. این به دلیل کمبود تعداد پیکسلهای متناوب برای اعمال الگوریتم دیترینگ است. به عنوان مثال، با استفاده از ماتریس ۳ * ۳، ممکن است به دلیل تناوب پیکسلها، الگوریتم به راحتی نتواند تغییرات رنگی را به خوبی پخش کند و باعث ایجاد نویز در تصویر شود.

٠٢

```
import numpy as np
```

```
def create_dither_matrix(m):
    dither_matrix = np.zeros((m, m), dtype=int)
    for i in range(m):
        for j in range(m):
        dither_matrix[i, j] = (i * m + j) * 256 / (m * m) - 1
    return dither_matrix

# * * * مثال: ساخت ماتریس دیترینگ * m = 4

dither_matrix = create_dither_matrix(m)

print(dither_matrix)
```

۳ در الگوریتم Floyd-Steinberg ، دلیلی که ۵ درایه ماتریس توزیع برابر با صفر است، این است که میخواهیم از اطرافیان پیکسل فعلی کمک بگیریم تا خطای رنگی را پخش کنیم. یعنی اگر ما از یک جهت بیشتر فشردهسازی کنیم، میخواهیم کمکی از جهت مقابل داشته باشیم و بر عکس. به این ترتیب، مقدار ۵ (یعنی پنجمین درایه) در ماتریس، برای کمک به پخش خطای رنگی از پیکسل فعلی استفاده نمیشود و بنابراین برابر با صفر قرار داده میشود.

۴. تابع find_closest_palette_color در الگوریتمهای مانند Floyd-Steinberg و ordered dithering برای انتخاب بهترین رنگ از پالت رنگهای مجاور به رنگ فعلی (معمولاً به عنوان رنگ برای پیکسل جدید) استفاده میشود. این تابع با محاسبه فاصله رنگها در فضای رنگی (معمولاً RGB) و انتخاب نزدیک ترین رنگ از پالت، کمک میکند تا خطای رنگی در هنگام تبدیل تصاویر کمتر شود و تصویر نهایی با کیفیت بهتری ایجاد شود.