

تمرین سری دوم سیستم های چندرسانه ای

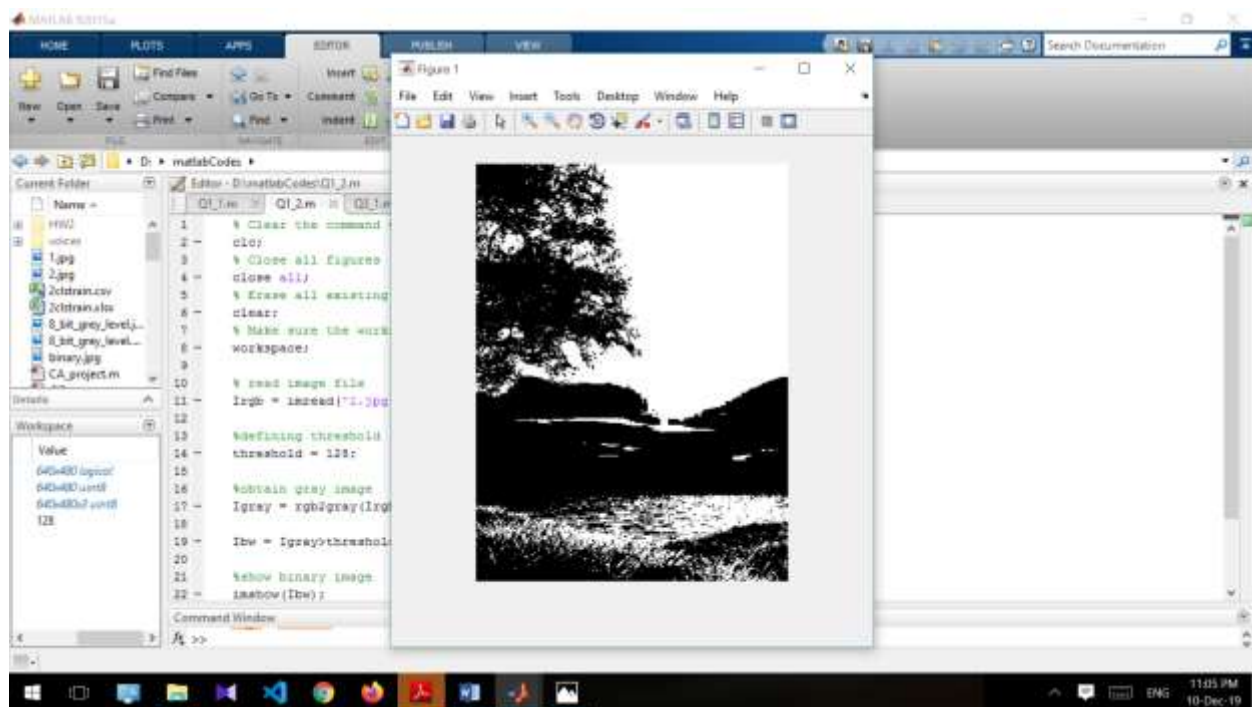
فاطمه غلام زاده

۹۵۳۱۰۶۰

سوال ۱-۱ : کد این سوال ضمیمه شده است. در زیر تصویر اصلی و تصویر نتیجه نشان داده شده :



سوال ۱-۱ : کد این سوال ضمیمه شده است. در زیر تصویر نتیجه نشان داده شده :



سوال ۱-۲:

دیترینگ روشی برای کاهش اثر banding هنگام کاهش رزولوشن رنگ است. این روش با کاهش جزئیات اطلاعات مکانی تصویر، کاهش اثر رزولوشن را جبران می کند.

الگوریتم فلوید اشتاینبرگ یک الگوریتم دیترینگ است. این الگوریتم با استفاده از انتشار خطا به پیکسل های مجاور اثر کاهش رزولوشن را از بین می برد .

سوال ۲-۲:

سوال ۲.۲:

$$\begin{bmatrix} 100 & 200 & 10 \\ 0 & 210 & 20 \\ 90 & 0 & 0 \end{bmatrix}$$

$$\text{oldpixel} = p[0][0] = 100 > 127 \rightarrow 0$$

$$\text{error} = 100$$

$$p[1][0] = \frac{v}{14} \times \text{error} = \frac{v}{14} \times 100 = 70.9 \approx 70$$

$$p[0][1] = \frac{u}{14} \times \text{error} + \text{oldpixel} = 13 + 200 = 213$$

$$p[1][1] = \frac{1}{14} \times \text{error} + \text{oldpixel} = 7 + 210 = 217$$

$$\Rightarrow \begin{bmatrix} 0 & 213 & 10 \\ 70 & 217 & 20 \\ 90 & 0 & 0 \end{bmatrix}$$

$$\text{oldpixel}[1][0] = 84 < 128 \rightarrow \text{newpixel}[1][0] = 0$$

$$\text{error} = 84 - 0 = 84$$

$$P[1][0] = 90 + \frac{V}{14} \times 84 = 110$$

$$P[0][1] = 211 + \frac{U}{14} \times 84 = 294,4 \approx 294$$

$$P[1][1] = 222 + \frac{A}{14} \times 84 = 234$$

$$P[2][1] = 0 + \frac{1}{14} \times 84 = 6,1 \approx 6$$

$$\begin{bmatrix} 0 & 195 & 10 \\ 0 & 234 & 20 \\ 110 & 23 & 0 \end{bmatrix}$$

$$\text{oldpixel}[1][0] = 110 < 128 \rightarrow \text{newpixel}[1][0] = 0$$

$$\text{error} = 110 - 0 = 110$$

$$P[1][1] = 234 + \frac{U}{14} \times 110 = 255$$

$$P[2][1] = 6 + \frac{A}{14} \times 110 = 11,9 \approx 12$$

$$\begin{bmatrix} 0 & 295 & 10 \\ 0 & 255 & 20 \\ 0 & 12 & . \end{bmatrix}$$

$$\text{oldpixel}[1][0] = 294 > 128 \rightarrow \text{newpixel}[1][0] = 200$$

$$\text{error} = 294 - 200 = 94$$

$$P[1][1] = 255 + \frac{V}{14} \times 94 = 260$$

$$P[0][2] = 10 + \frac{A}{14} \times 94 = 18$$

$$P[1][2] = 20 + \frac{1}{14} \times 94 = 26$$

$$\begin{bmatrix} 0 & 200 & 18 \\ 0 & 260 & 26 \\ 0 & 12 & . \end{bmatrix}$$

$$\text{oldpixel}[1][1] = 260 > 128 \rightarrow \text{newpixel}[1][1] = 200$$

$$\text{error} = 260 - 200 = 60$$

$$P[2][1] = 12 + \frac{V}{14} \times 60 = 24$$

$$P[0][2] = 18 + \frac{U}{14} \times 60 = 32$$

$$P[1][2] = 26 + \frac{A}{14} \times 60 = 34$$

$$P[2][2] = 0 + \frac{1}{14} \times 60 = 4$$

$$\begin{bmatrix} 0 & 200 & 32 \\ 0 & 200 & 34 \\ 0 & 24 & 4 \end{bmatrix}$$

$$\text{oldpixel}[r][1] = 9V < 12V \rightarrow \text{newpixel} = 0$$

$$\text{error} = 9V - 0 = 9V$$

$$P[r][1] = 43 + \frac{1}{14} \times 9V = 41$$

$$P[r][r] = 1 + \frac{0}{14} \times 9V = 1$$

$$\begin{bmatrix} 0 & 255 & 32 \\ 0 & 255 & 11 \\ 0 & 0 & 31 \end{bmatrix}$$

$$\text{oldpixel}[0][r] = 32 < 12V \rightarrow \text{newpixel}[0][r] = 0$$

$$\text{error} = 32 - 0 = 32$$

$$P[1][r] = 11 + \frac{1}{14} \times 32 = 9$$

$$\begin{bmatrix} 0 & 255 & 0 \\ 0 & 255 & 9 \\ 0 & 0 & 31 \end{bmatrix}$$

$$\text{oldpixel}[1][r] = 9 < 12V \rightarrow \text{newpixel}[1][r] = 0$$

$$\text{error} = 9 - 0 = 9$$

$$P[r][r] = 31 + \frac{1}{14} \times 9 = 32$$

$$\begin{bmatrix} 0 & 255 & 0 \\ 0 & 255 & 0 \\ 0 & 0 & 32 \end{bmatrix}$$

$$\text{oldpixel}[r][r] = 32 < 0 \rightarrow \text{newpixel}[r][r] = 0$$

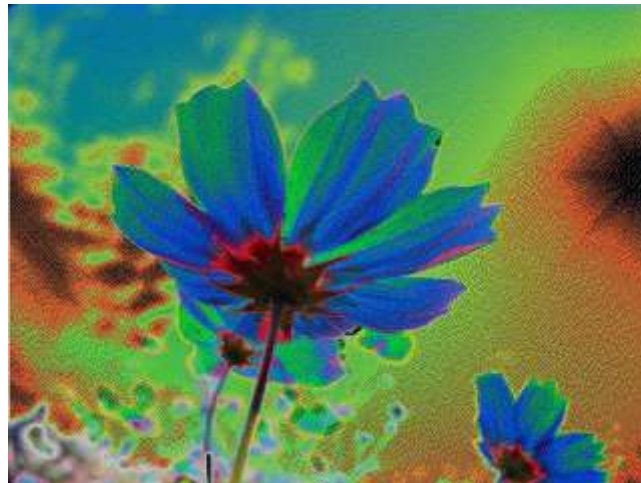
$$\Rightarrow \begin{bmatrix} 0 & 255 & 0 \\ 0 & 255 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

سوال ۲-۳: کد این قسمت ضمیمه شده است.

تصویر اصلی:



تصویر با خطا :



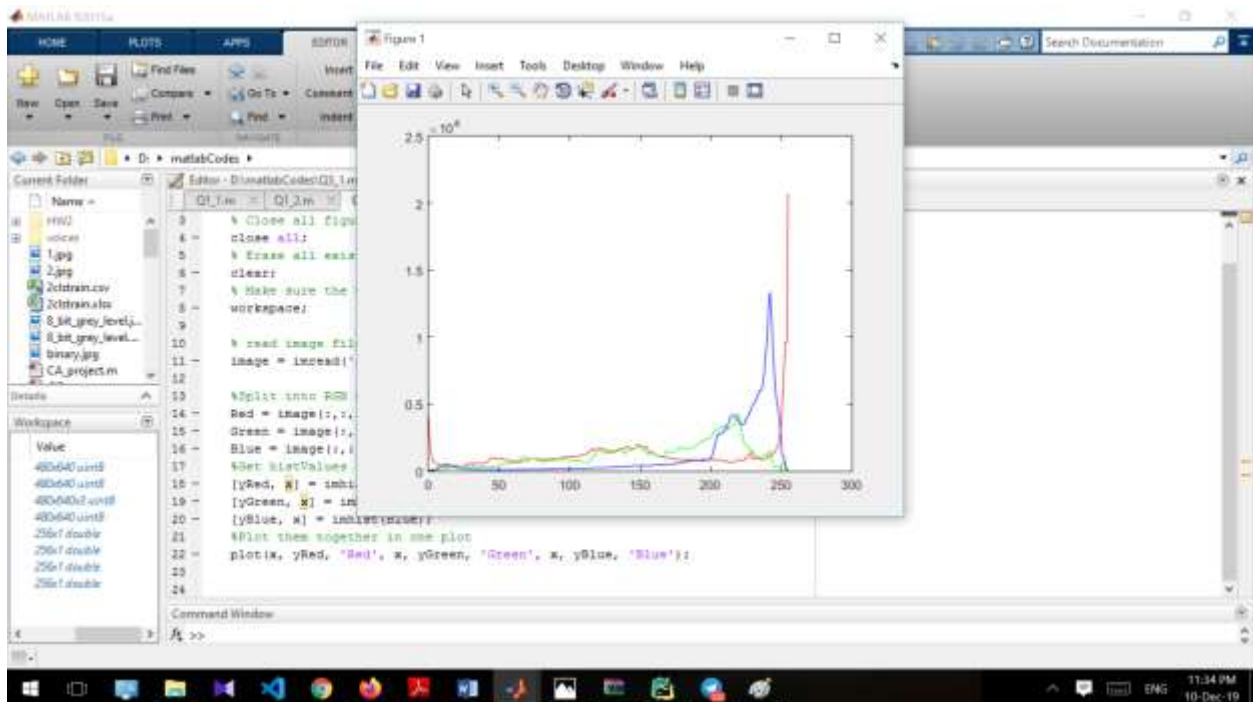
تصویر بدون خطا :



سوال ۲-۴ :

این الگوریتم دیترینگ را با انتشار خطا انجام می دهد. اختلاف مقدار جدید با مقدار قبلی را حساب می کنیم و این اختلاف را با ضربایی مشخص به پیکسل های مجاور انتقال می دهیم. مقدار خطا به ما نشان می دهد که مقدار واقعی پیکسل با مقدار کنونی آن چقدر متفاوت است و با اضافه کرن این مقدار به پیکسل های مجاور سعی می کنیم که میزان شدت کم شدن آن ها را کاهش بدهیم .

سوال 1-3 : کد این قسمت به زبان متلب ضمیمه شده. هیستوگرام :



۲-3 کد این سوال ضمیمه شده است. (به زبان پایتون)

سوال ۴ :

PNG files

Choose PNG when you need a small file that maintains its original quality. PNG files support millions of colors, plus varying degrees of transparency — so they're perfect for graphic image files, like logos, charts, and infographics.

PNGs maintain their original quality when compressed. When you need to make a PNG smaller, it will look as clear as the original file. Note that PNGs are typically larger files than JPEGs and GIFs, and the larger the file size, the longer it takes to load on the web

JPEG files

JPEG is the go-to format for online photos. It supports a full spectrum of colors, and almost all devices and programs can open and save to the JPEG format — making it the most universal of the four. JPEG files are ideal when you want to keep file size down and don't mind giving up a little quality to create a very small file. That said, JPEG quality drops when images are edited and saved. If you plan to continually edit your files, JPEG is not the format for the job. But if you need to display photos online, JPEGs are just right.

In addition to losing quality in compression, JPEGs don't support transparency. So, don't use JPEGs for line-based graphics, especially over another image or background color.

GIF files

Use GIF for simple web graphics with limited colors. GIF files are the smallest of the four because they are always reduced to 256 colors, making for fast-loading visuals. That said, GIF files aren't recommended for files with a large range of colors, like photographs or other detailed imagery. But if you're working with small graphics, like banners, charts, and buttons, GIF is the best format for the job.