



گزارش تکلیف پردازش تصویر سری ۱

امیرحسین دارایی

۹۷۳۳۰۲۳

سوال ۱

سوالات ۷ و ۹ را از فصل دوم کتاب گنزالز (ویرایش سوم) حل کنید.

- 2.7 Suppose that a flat area with center at (x_0, y_0) is illuminated by a light source with intensity distribution

$$i(x, y) = Ke^{-[(x-x_0)^2 + (y-y_0)^2]}$$

Assume for simplicity that the reflectance of the area is constant and equal to 1.0, and let $K = 255$. If the resulting image is digitized with k bits of intensity resolution, and the eye can detect an abrupt change of eight shades of intensity between adjacent pixels, what value of k will cause visible false contouring?

جواب ۷:

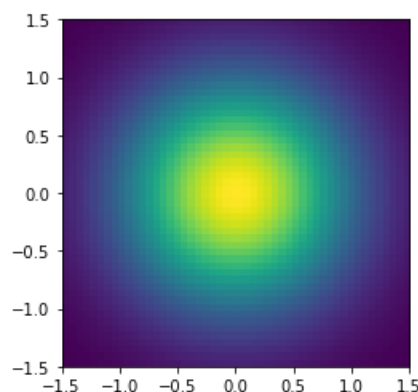
پاسخ سوال ۷:

$$f = i \times r = 255 \exp\left\{-[(x-x_0)^2 + (y-y_0)^2]\right\}$$

تصویر: $f(x, y)$
شدت نور: $i(x, y)$
بازتاب نور: $r(x, y)$

برای ساده‌تر
 $x_0 = 0, y_0 = 0$: $f(x, y) = 255 e^{-[x^2 + y^2]}$

تصویر آن به شکل زیر است:



$$\frac{255-0+1}{A} = 8 \Rightarrow A = 32 \quad \text{۳۲ بایت متوسط شدت فرکانس است.}$$

$$32 = 2^x \Rightarrow x = 5 \text{ bits} \quad \checkmark$$

★ 2.9 A common measure of transmission for digital data is the *baud rate*, defined as the number of bits transmitted per second. Generally, transmission is accomplished

in packets consisting of a start bit, a byte (8 bits) of information, and a stop bit. Using these facts, answer the following:

- (a) How many minutes would it take to transmit a 1024×1024 image with 256 intensity levels using a 56K baud modem?
- (b) What would the time be at 3000K baud, a representative medium speed of a phone DSL (Digital Subscriber Line) connection?

جواب ۹:

Baud rate: # of bits per second

پایه سوال ۹: 1 packet: 10 bits (start bit, 8 bits, stop bit)

دیتای ارسالی بدون در نظر گرفتن بیت $1024 \times 1024 = 2^{10} \text{ pixel} \times 2^8 \text{ bit} = 2^{18}$

تعداد بیت های ارسالی بدون در نظر گرفتن بیت $2^{18} \div 2^3 = 2^{15}$

دیتای ارسالی با در نظر گرفتن بیت $2^{15} \times \frac{10 \text{ بیت}}{1 \text{ پکت}} = 2^{15} \times 10 \text{ bits}$

زمان $\frac{2^{15} \times 10}{56000} = 157.24 \text{ ثانیه}$ معادل ۲،۵۷ دقیقه ✓

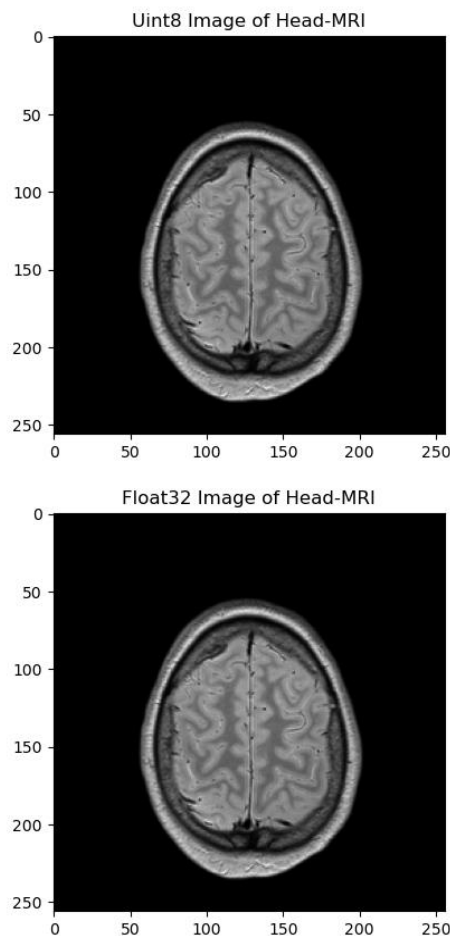
b) $v = 3000 \times 10^3 \Rightarrow$ زمان $\frac{2^{15} \times 10}{3000 \times 10^3} = 3.57 \text{ ثانیه}$ معادل ۰،۰۰۳۵۷ دقیقه ✓

سوال ۲

آ تصویر Head-MRI.tif را به صورت خاکستری بخوانید. سپس یک نمودار ۲ در ۱ بسازید و در قسمت بالا تصویر uint8 را نمایش دهید و در قسمت پایین، ابتدا نوع تصویر را به float تغییر داده و سپس تصویر را با تقسیم بر ۲۵۵ به بازه ۰ تا ۱ برده و با تنظیم vmin, vmax آن را نشان دهید. برای تصاویر عنوان مناسب قرار دهید. (۲۰٪)

جواب: ابتدا تصویر را بصورت خاکستری میخوانم. از آن جایی که اطلاعات هر پیکسل از تصویر بصورت uint8 و از کلاس نامپای می باشد آن ها را با استفاده از تابعی موجود در کتابخانه نامپای بصورت float32 ذخیره میکنم. در ادامه بر ۲۵۵ تقسیم میکنم و آن را نمایش می دهم.

خروجی برنامه:



تصویر ۱

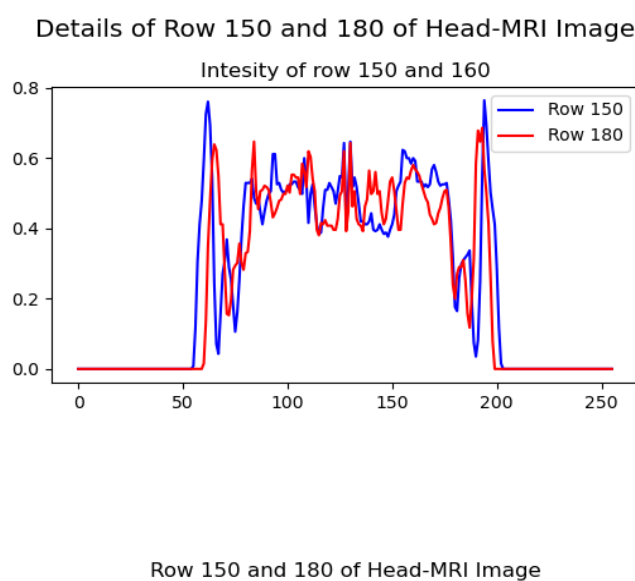
ب) دو ردیف ۱۵۰ و ۱۸۰ تصویر را در دو متغیر مختلف بریزید و با استفاده از کتابخانه matplotlib یک نمودار رسم کنید و با استفاده از دستور subplot/subplots آن نمودار را به دو قسمت تقسیم کنید. در قسمت بالا نمودارهای شدت ردیف‌های داده شده را با استفاده از دستور plot رسم کنید (ردیف ۱۵۰ با رنگ آبی و ردیف ۱۸۰ با رنگ قرمز) همچنین legend را نیز نشان دهید و تیتیر مناسب بگذارید. (۶۰٪)

ج) در قسمت پایین دو ردیف بالا را به صورت تصویر نمایش دهید و محورها را غیر فعال کنید و عنوان مناسب بگذارید. (دقت داشته باشید که تصویر شما خاکستری است و باید هنگام نمایش نیز به صورت خاکستری نمایش داده شود). (۲۰٪)

جواب: برای نوشتن این برنامه باید ابتدا سطرهای ۱۵۰ و ۱۸۰ را از تصویر استخراج کنیم. برای این کار باید به این نکته توجه کنیم که شمارش از صفر شروع میشود و سطر ۱۵۰ شماره ۱۴۹ را دارد. این سطرها را در دو متغیر متفاوت ذخیره کردم و به سادگی آن‌ها را رسم کردم.

برای بخش (ج) باید ردیف‌های انتخاب شده را نشان دهیم که با رسم شدت تفاوت دارد. برای به تصویر کشیدن این ردیف‌ها یک بعد به ارایه اضافه کردم تا آرایه قابل به تصویر کشیدن باشد. سپس برای این که با دستور کار منطبق باشم، دو سطر انتخابی را با هم stack کردم تا بتوانم آن‌ها را با یکدیگر در یک subplot نمایش دهم.

خروجی:



سوال ۳

(در این سوال تنها می‌توانید از کتابخانه‌های OpenCV, Numpy استفاده کنید.)

(آ) تصویر MRI-Head.png را به صورت خاکستری بخوانید. (۵٪)

(ب) ویدیوی MRI.avi را بخوانید. فریم اول تصویر را بخوانید و آن را از تصویر دریافتی در قسمت الف کم کنید تا نویز تصویر به دست آید. سپس میانگین و انحراف معیار نویز را بیابید. از تمام فریم‌های ویدیو میانگین گرفته و در خروجی نمایش دهید و تصویر نهایی را ذخیره کنید. (راهنمایی: برای بدست آوردن نویز می‌توانید از نوع داده int32 استفاده کنید). (۶۰٪)

(ج) تصویر خروجی قسمت قبل را از تصویر قسمت الف کم کنید و میانگین و انحراف معیار نویز تصویر نهایی را بیابید و با روابط تدریس‌شده انطباق دهید. (۱۵٪)

(د) تصویر mask.png را بخوانید و با استفاده از عملیات منطقی مناسب، نواحی مشخص شده را در تصویر به دست آمده در قسمت الف را نشان دهد. تصویر را در خروجی نشان دهید. (۲۰٪)

جواب: الف و ب و ج) برای این برنامه ابتدا تصویر را می‌خوانم. سپس ویدیو را بارگیری کرده و فریم اول را از ویدیو می‌خوانیم. این فریم ۳ کانال دارد و آن را به یک کانال یعنی Grayscale تبدیل کردم. سپس تصویر لود شده را از فریمی که ذخیره کرده ام، کم کردم و به آن نویز می‌گوییم. میانگین و انحراف معیار نویز را با توابع کتابخانه نامپای می‌توان حساب کرد. در نهایت فریم اول، تصویر اصلی و نویز حساب شده را نمایش می‌دهم. با مشاهده خروجی میتوان مطالبی که سرکلاس تدریش شد را به وضوح مشاهده کرد: میانگین و انحراف معیار نویز در حالت ثانویه (زمانی که تصاویر ویدیو میانگین گیری شده اند) بسیار کمتر از حالت اولیه (زمانی که فقط فریم اول ویدیو انتخاب شده است) می‌باشد.

(د) در این قسمت ابتدا تصویر ماسک را لود میکنیم و سپس آن را روی تصویر اصلی بصورت فرآیند شرطی AND اعمال میکنیم و در انتها نمایش می‌دهیم.

خروجی:

در صفحه بعد مشاهده کنید.

Noise

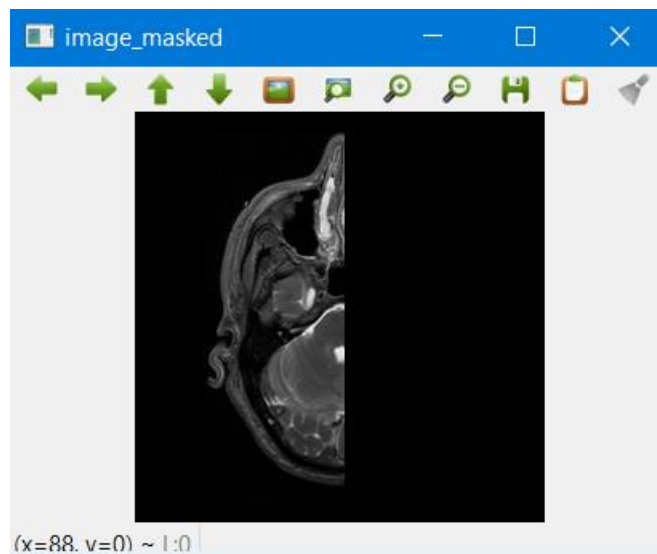
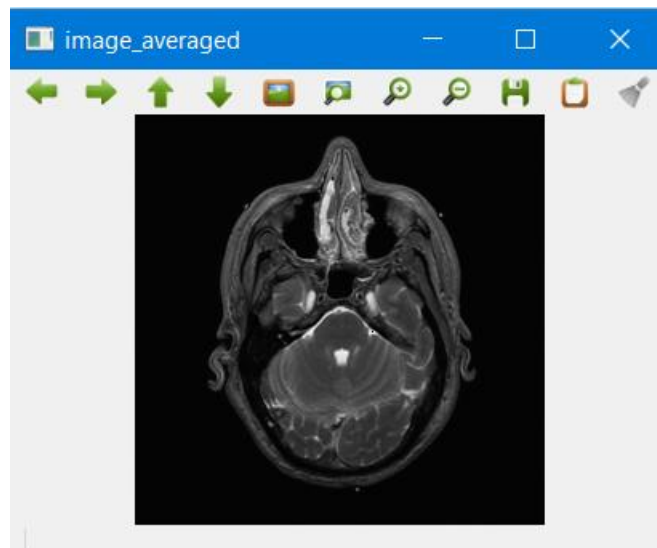
min:-52 max:38 shape:(256, 256):

mean:0.8322601318359375 std:8.113755210440392 type:<class 'numpy.int16'>

 noise2

min:-10.85 max:5.066 shape:(256, 256)

mean:0.73191845703125 std:2.5521150444558436 type:<class 'numpy.float64'>



تصویر را در خروجی به دست آورید. (

خروجی ۱:

_____ Color Image _____

min:0 max:255 shape:(366, 409, 3)

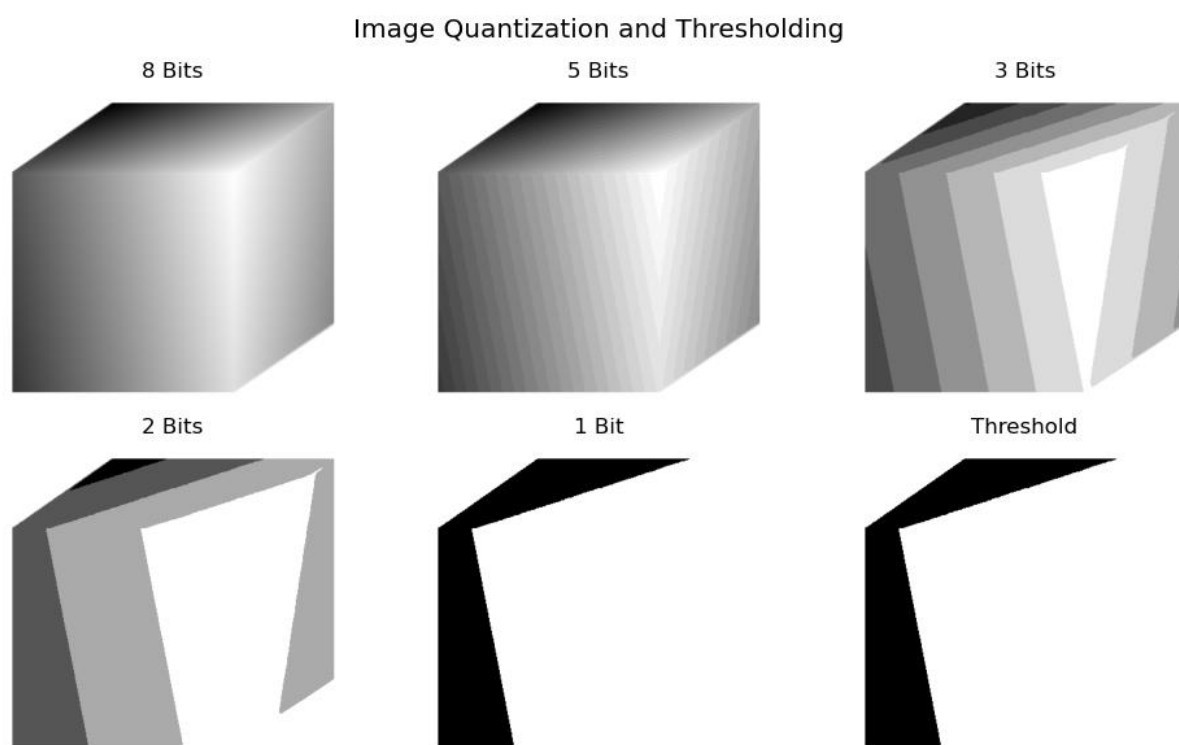
mean:189.1512329596822 std:83.1740414164082 type:<class 'numpy.uint8'>

_____ Image Grayscale _____

min:29 max:255 shape:(366, 409)

mean:189.84672064344596 std:53.9358216339193 type:<class 'numpy.uint8'>

خروجی ۲:



سوال ۵

(آ) تصویر AUT-DIP.png را به صورت خاکستری بخوانید و آن را به ۶ قسمت مساوی تقسیم کنید (به نحوی که هر یک از حروف در مرکز آن قسمت بیافتد) (۱۰٪)

(ب) حرف اول را با استفاده از تبدیل scaling با ضریب ۲ بزرگنمایی کنید. و سپس آن را به نحوی برش بزنید (cropping) که ابعاد آن از ابعاد حالت اولیه بیشتر نشود. (فقط حرف بزرگ شود) (۱۰٪)

(ج) حرف دوم تبدیل کج شدگی افقی (Horizontal Shear) را با ضریب ۰.۲ اعمال کنید. (۱۰٪)

(د) حرف سوم را با استفاده از تبدیل جابجایی (Translation) به میزان ۱۰۰ پیکسل به پایین و ۸۰ پیکسل به چپ تبدیل کنید. (۱۰٪)

(ه) حرف چهارم را با استفاده از اطلاعاتی که درون درس مبنی بر چرخش با روش مستقیم حول نقطه (۰, ۰) به میزان ۲۵ درجه بچرخانید. (۲۰٪)

(و) حرف پنجم را مانند قسمت ث با استفاده از چرخش با روش معکوس به میزان ۲۵- درجه بچرخانید. (۲۰٪)

(ز) حرف آخر را با استفاده از توابع آماده موجود حول نقطه‌ی مرکزی تصویر به اندازه ۴۵ درجه بچرخانید. تابع تبدیل آن را در خروجی چاپ کنید و به صورت کلی تحلیل خود از این تابع آماده در گزارش بنویسید. (۱۰٪)

(ح) حال تمام ۶ تصویر هم اندازه به دست آمده را با همان ترتیب تصویر خوانده شده به هم بچسبانید و با استفاده از کتابخانه Matplotlib و با توجه به ملاحظات گفته شده در سوال قبل (خاموش کردن محور ها و افزودن عنوان و ...) چاپ کنید. (۱۰٪)

تصویر را به صورت خاکستری خواندم و آن را به شش قسمت مساوی تقسیم کردم به این صورت که ابتدا طول و عرض تصویر را بدست آورم و با تقسیم کردم طول بر ۳ و عرض بر ۲ و استفاده از selection آرایه تصویر آن را به شش تصویر تقسیم کردم بطوری که اکنون هر یک از حروف در مرکز آن قسمت افتاده است.

حرف اول را با استفاده از تابع resize یا تبدیل scaling به اندازه ۲ برابر بزرگنمایی کردم. ضابطه تبدیل بزرگنمایی به این صورت است:

$$T = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

و سپس آن را با دانستن طول و عرض تصویر جدید و انتخاب کردن مرکز آن بصورت آرایه ای برش زدم و ابعاد برش را طوری باید تنظیم می کردم که ابعاد تصویر نهایی برابر ابعاد تصویر اولیه شود.

در مرحله سوم، به حرف دوم تبدیل کج شدگی افقی را با ضریب ۰.۲ اعمال کردم. تابع تبدیل آن بصورت زیر است و آن را با استفاده از یکی از توابع کتابخانه opencv به تصویر اعمال کردم:

$$T = \begin{pmatrix} 1 & 0.2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

در مرحله چهارم حرف سوم را با استفاده از تبدیل جابجایی به میزان ۱۰۰ پیکسل به پایین و ۸۰ پیکسل به چپ انتقال دادم. تابع تبدیل آن بصورت زیر است و آن را با استفاده از یکی از توابع کتابخانه opencv به تصویر اعمال کردم:

$$T = \begin{pmatrix} 1 & 0 & -80 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{pmatrix}$$

در مرحله پنجم حرف چهارم را با روش مستقیم به اندازه ۲۵ درجه چرخاندم. تابع تبدیل آن بصورت زیر است و آن را با استفاده از یکی از توابع کتابخانه opencv به تصویر اعمال کردم:

$$T = \begin{pmatrix} \cos 25 & -\sin 25 & 0 \\ \sin 25 & \cos 25 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

در مرحله ششم به شکل بالا ولی به میزان منفی تصویر را چرخاندم.

در مرحله هفتم حرف ششم را با استفاده از توابع آماده موجود به اندازه ۴۵ درجه چرخاندم. تابع تبدیل آن به صورت زیر است:

$$\begin{bmatrix} 0.70710678 & 0.70710678 & 0 \\ 0.70710678 & -0.70710678 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.70710678 & 0.70710678 & 0 \\ 0.70710678 & -0.70710678 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

تحلیل من این است که این تابع بجای نوشتن ماتریس تبدیل از ابتدا، فقط با گرفتن زاویه چرخش، ماتریس تبدیل را تولید می‌کند و لازم به پیاده سازی تابع تبدیل نیست و همچنین این تابع به تصویر نهایی یک transform نیز اعمال می‌کند تا وسط بودن تصویر محفوظ بماند و تغییر نکند.

در مرحله هشتم (مرحله آخر) تمام تصویر های تولید شده را به صورت معکوس مرحله اول به هم چسباندم و تصویر (ماتریس) خروجی را نمایش و ذخیره سازی کردم و خروجی به شکل زیر است:

Final Image

