



به نام او



دانشگاه صنعتی امیرکبیر

# پردازش تصویر

## آزمون پایانی

زمان ارسال: ۹۹/۱۱/۱۴

مهلت تحویل: ۹۹/۱۱/۲۱

استاد درس: دکتر حامد آذرنوش

تدریس یاران:

ارمغان سرور

حسین قاسم دامغانی

نیم سال پاییز ۹۹-۰۰

## توضیحات مهم

- اطلاعات در قالب کد و گزارش باید تحویل داده شوند؛ گزارش مناسب در کنار کد درست قابل قبول است. ۷۰٪ نمره‌ی تمارین کد نویسی شما را خود کد تشکیل می‌دهد و ۳۰٪ باقی‌مانده از ضرب نمره‌ی کد شما در نمره‌ی گزارش‌تان تشکیل خواهد شد. برای مثال اگر کد شما ۸۰٪، نمره را دریافت کند و گزارش‌تان ۱۰٪، نمره‌ی تمرین شما  $۵۸/۴ = ۳۰\% \times ۱۰\% + ۸۰\% \times ۷۰\%$  خواهد بود، پس به نکات ذکر شده در مورد گزارش نویسی توجه فرمایید.
- در کد سعی کنید کامنت مناسب بگذارید. در گزارش، ورودی‌ها و خروجی‌ها را با توضیحات تکمیلی قرار دهید. از قرار دادن کد در گزارش خودداری کنید.
- در گزارش، برای پاسخ به هر سوال از ابتدای صفحه‌ی جدید استفاده کنید.
- کد مربوط به هر سوال و فایل‌های مرتبط با آن را در پوشه‌های جداگانه قرار دهید و طوری تنظیم شود که بدون هیچ مشکلی اجرا شود.
- لطفاً برنامه‌ریزی بفرمایید تا قبل از موعد مقرر ارسال بفرمایید، مهلت ارسال آزمون تمدید نخواهد شد.
- این آزمون شامل سهمیه‌ی تأخیر نیست و تأخیر در ارسال آزمون معادل از دست رفتن کل نمره‌ی آزمون خواهد بود.
- هرگونه مشابهت در کدها و گزارش منجر به از دست رفتن کل نمره‌ی مسأله خواهد شد.
- سوالات درسی خود را فقط از طریق ایمیل [dipautfall2020@gmail.com](mailto:dipautfall2020@gmail.com) مطرح بفرمایید و عنوان مناسب شامل شماره‌ی سوال نیز قرار دهید.

- نحوه‌ی ارسال: فایل گزارش را به همراه کدهای نوشته شده در قالب یک فایل فشرده‌ی zip به اسم FinalExam\_Num باشد که Num شماره‌ی دانشجویی شما هست، مانند FinalExam\_9433001. فقط از طریق سامانه‌ی مدیریت یادگیری Moodle ارسال بفرمایید.

۱ ۳۰٪

تصویر خاکستری و نویزدار “noisy\_skull.png” را بخوانید. یک الگوی ثابت از نویز در تمام تصویر موجود است. مراحل زیر شما را به سمت هدف مسأله هدایت خواهند کرد:

آ) نمودار فراوانی نسبی<sup>۱</sup>، میانگین و واریانس شدت‌های نویز موجود در آن را استخراج کنید، نمایش دهید و روش کار را شرح دهید.

ب) در گزارش، جنس تابع چگالی احتمال<sup>۲</sup> نویز را با توجه به توزیع‌های مهم آماری حدس بزنید.

ج) فیلتر تطبیقی محلی کاهش نویز<sup>۳</sup> را با پنجره‌ی  $9 \times 9$  بر روی تصویر پیاده کنید و تصویر ۸ بیتی تمیز شده را بدست آورید. حاشیه‌سازی دور تصویر<sup>۴</sup>، `cv.BORDER_REPLICATE` و به تعداد لازم باشد.

د) تصویر اولیه را به روش میانه‌گیری با پنجره‌ی  $7 \times 7$  نیز فیلتر کنید.

ه) در یک نمودار ۳ در ۲، تصویر اصلی و تصاویر فیلتر شده از دو روش را در ستون‌های مجزا نمایش دهید و در زیر آن‌ها، نمودار فراوانی نسبی نویز آن‌ها را نمایش دهید.

و) با مقایسه‌ی تصویر اصلی و فیلتر شده‌ی تطبیقی، آیا برای کاهش نویز بهبود حاصل شده یا خیر؟ دلیل را شرح دهید.

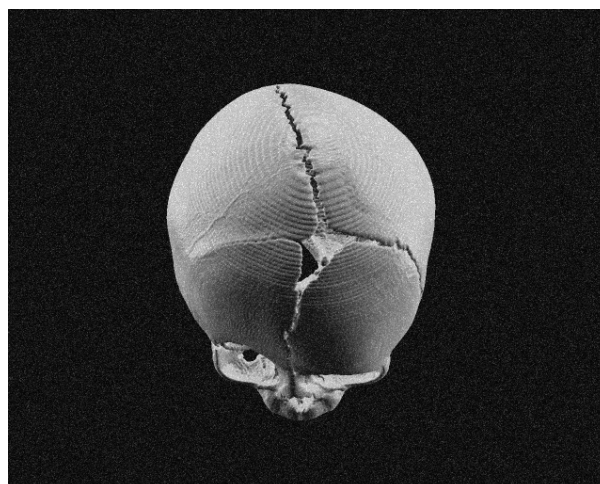
ز) آیا برای کاهش نویز، روش میانه‌گیری و فیلتر تطبیقی، نتایج مشابهی داشتند؟ با توجه به خروجی تصویر هر دو روش، مزیت فیلتر تطبیقی را از نظر کلی بیان کنید. توضیحات بیشتر و راهنمایی:

I) کلیت عملیات فیلتر تطبیقی باید توسط شما نوشته بشود. در باقی عملیات‌ها و جزئیات فیلتر تطبیقی، توابع آماده‌ی `numpy`، `matplotlib` و `opencv` مجازند.

II) در پیاده‌سازی فیلتر تطبیقی، از روش برش آرایه‌ها در numpy استفاده کنید تا لازم نباشد به شکل دستی تک‌تک ۸۱ پیکسل هر پنجره را استخراج کنید!

III) برای مقایسه‌ی درست بین نمودارهای فراوانی نسبی، محدوده‌ی عمودی رسم نمودارها را یکسان تنظیم کنید. `matplotlib.pyplot.ylim`

IV) برای محاسبه و نمایش فراوانی هر دسته، دستورالعمل توابع را با دقت بخوانید، لبه‌های دسته‌ها به درستی انتخاب شوند و گزینه‌ی مناسب برای نسبی شدن فراوانی، انتخاب شود. `numpy.histogram` `matplotlib.pyplot.hist`



<sup>۱</sup> Normalized Histogram

<sup>۲</sup> PDF: Probability Density Function

<sup>۳</sup> Adaptive local noise reduction filter

<sup>۴</sup> padding

۲ ۲۰%

تصویر خاکستری “reflections.jpg” را بخوانید. برنامه‌ای بنویسید که با هر بار کلیک کاربر، مختصات آن را دریافت کرده و عملیات پرکردن حفره<sup>۵</sup> را روی آن با نقطه‌ی اولیه‌ی داده شده و کرنل ۳ در ۳ به شکل +، پیاده کند و نتیجه‌ی جدید را نمایش دهد. اگر کاربر دکمه‌ی ESC (با کد ASCII ۲۷) را فشار داد، برنامه تمام شود.

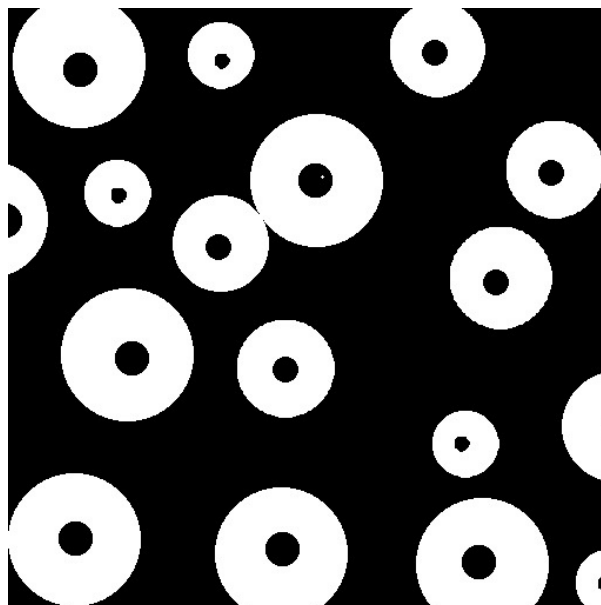
توضیح و راهنمایی: هر بار که کاربر کلیک می‌کند، باید همان تصویر پردازش شود، نباید تصویر اولیه از اول پردازش شود، به عبارت دیگر در این برنامه باید توانست با کلیک‌های پیاپی، کل صفحه را سفید کرد. بدین منظور در اولین خط هر تابعی که تعریف می‌کنید، عبارت زیر را قرار دهید تا تصویر با نام img یک متغیر سراسری به حساب بیاید و در داخل بدنه‌ی توابع بتوانید آن را تغییر بدهید:

```
global img
```

مثال یک برنامه‌ی فرضی:

```
x = 10
def bar():
    global x
    x += 1
    print(x)
bar()
```

بدون استفاده از global این برنامه خطای UnboundLocalError نمایش می‌دهد.



۳ ۲۰%

یکی از تکنیک‌های تشخیص اجسام در حال حرکت، استفاده از روش کم کردن پس‌زمینه تصویر است؛ بدین صورت که پس‌زمینه‌ی فیلم ورودی که انتظار داریم تقریباً ثابت باشد با میانگین‌گیری از فریم‌های فیلم بدست می‌آید و سپس با کم کردن آن از هر فریم، جسم در حال حرکت مشخص می‌گردد. در این سوال شما میبایست روش مشابهی را به کار بگیرید و اجسام در حال حرکت در فیلم "Q\_three.avi" را با رنگی کردن (به طور خاص صفر کردن کانال‌های رنگی سبز و آبی) شناسایی نمایید.

نکته: برای سادگی کار، رنگ پس‌زمینه را از RGB به خاکستری تغییر داده و با کنار هم گذاشتن فریم‌های بدست آمده، خروجی را در یک فایل جدید ذخیره کرده و به طور همزمان با اجرای برنامه نمایش دهید. به عنوان نمونه نیز یک فریم را در گزارش نهایی قرار دهید. شکل زیر مبنای هدف این سوال را نشان می‌دهد.



۴ ۳۰٪

از روش‌های بخش‌بندی تصاویر که مبتنی بر عملیات بر روی نواحی مختلف تصویر است، می‌توان به الگوریتم region-based growing که در درس با آن آشنا شده اید اشاره نمود. در این سوال شما باید این الگوریتم را بر روی تصویر "fMRI.jpg" پیاده‌سازی نمایید.

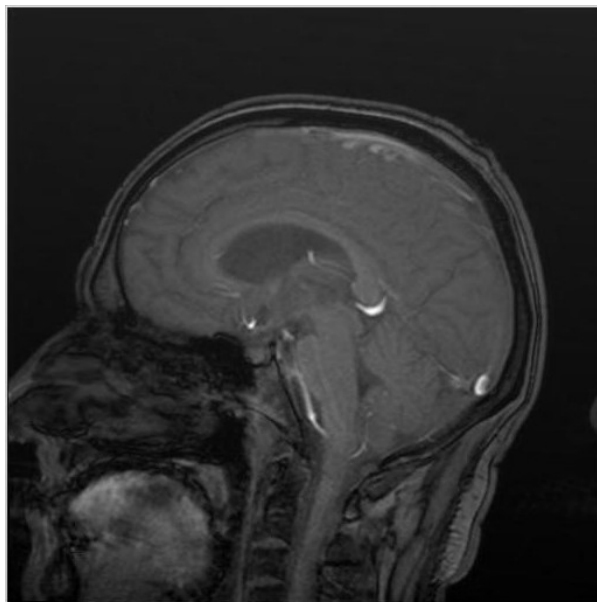
بدین منظور:

آ) برنامه پس از اجرا شدن باید منتظر وارد کردن یک دانه‌ی اولیه (seed) توسط کاربر با کلیک کردن بماند.

ب) مقدار عددی آستانه (Threshold) استفاده شده در الگوریتم برابر با عدد ۱۱ قرار داده شود.

ج) هر دو همسایگی ۴ تایی و ۸ تایی باید در الگوریتم پیاده‌سازی شده در نظر گرفته شوند و باید با تعریف یک متغیر Boolean در برنامه، نوع همسایگی مشخص گردد. (برای مثال در همسایگی ۴ تایی باید جهات (۱ و ۰)، (۰ و ۱)، (۰ و -۱)، (-۱ و ۰) را مد نظر قرار دهید.)

د) پس از اتمام اجرای الگوریتم، تصویر بخش‌بندی شده به کاربر نمایش داده شود، خروجی همسایگی چهار با نام out4.png و همچنین خروجی همسایگی هشت با نام out8.png ذخیره گردد.



موفق باشید.