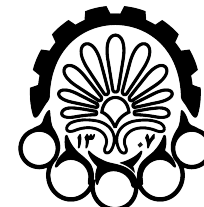


به نام او



دانشگاه صنعتی امیرکبیر
دانشکده‌ی مهندسی پزشکی
گروه بیوالکتریک



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

پردازش تصویر

تمرین شماره‌ی ۱
پایه‌های تصویر دیجیتال و عملیات‌های پایه

تاریخ ارسال: ۱۴۰۰/۱۱/۳۰

تاریخ نهایی تحویل: ۱۴۰۰/۱۲/۱۴

استاد درس:
دکتر حامد آذرنوش

تدریس‌یاران تمرین‌ها:

حمیدرضا ابوتی مهریزی

یلدا ظفری قدیم

امیرحسین شریفی صدر

نیم‌سال بهار ۱۴۰۰

۱ - ۲۵٪ تشریحی

(مختصات به شکل (x, y) و مطابق قرارداد درس است.)

آ) کوتاه‌ترین مسیر بین مختصات $(5, 0)$ تا $(0, 5)$ را با معیارهای D_4 و D_8 به شرطی که مقادیر مجاز مسیر $V = \{0, 1\}$ باشند بیابید. در صورت وجود چند راه‌حل، آن‌ها را بیان کنید. (۵۰٪)

$$\begin{bmatrix} 1 & 3 & 2 & 1 & 3 & 1 \\ 1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 3 & 0 & 1 \\ 1 & 2 & 3 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 3 \end{bmatrix}$$

ب) تابع درونیابی دوخطی نقاط بین مختصات $(2, 1)$ تا $(3, 2)$ را بدست آورید. (۵۰٪)

$$\begin{bmatrix} 19 & 13 & 6 & 1 \\ 29 & 0 & 7 & 2 \\ 10 & 22 & 17 & 18 \\ 31 & 27 & 22 & 23 \end{bmatrix}$$

توضیحاتی در مورد فرمت تصویری DICOM

نوع فایل (Digital Imaging and Communications in Medicine (DICOM با پسوند فایل dcm. (گاهی هم بدون پسوند) نوعی از فایل تصویری است که در تصویربرداری پزشکی استفاده می‌شود. برای خواندن این تصاویر در پایتون، باید بسته‌ی pydicom را نصب نمایید. راهنمای نصب آن در [این لینک](#) موجود است اما به شکل خلاصه همانند نصب numpy، دستور `pip install pydicom` باید اجرا شود و به راحتی روی دستگاه شما نصب می‌شود.

با خواندن این تصاویر به اطلاعات مفیدی می‌توانید دسترسی پیدا کنید؛ برای مثال خود آرایه‌ی تصویر، تاریخ و زمان تصویربرداری، تعداد بیت‌های اختصاص داده شده (برای هر پیکسل)، تعداد بیت‌های ذخیره شده (برای هر پیکسل)^۱، بخش از بدن که تصویربرداری شده، دسته‌بندی روش تصویربرداری (MRI, CT, etc.)، سن بیمار، کد شناسایی بیمار، جنسیت بیمار و اطلاعات لازم دیگر.

برای دسترسی به این اطلاعات، بعد از خواندن و ذخیره کردن فایل دایکام در یک متغیری مثلاً به نام ds، کافی است attribute مورد نظر را بخوانید.

```
>>> ds.Modality
'CT'
```

از این روش برای دستیابی به یک مورد خاص استفاده می‌شود، اما برای دیدن اطلاعات کلی فایل می‌توانید خود ds را چاپ کنید تا به فرم استاندارد خودش، جدولی از اطلاعات فایل را برایتان نمایش دهد.

```
>>> print(ds)
```

```
Dataset.file_meta -----
```

```
(0002, 0000) File Meta Information Group Length  UL: 194
```

```
(0002, 0001) File Meta Information Version      OB: b'\x00\x01'
```

```
(0002, 0002) Media Storage SOP Class UID        UI: CT Image Storage
```

```
(0002, 0003) Media Storage SOP Instance UID     UI: 1.3.6.1.4.1.14519.5.2.1.7777.9002.1798755185374
```

```
(0002, 0010) Transfer Syntax UID                UI: Implicit VR Little Endian
```

```
(0002, 0012) Implementation Class UID          UI: 1.2.40.0.13.1.1.1
```

```
(0002, 0013) Implementation Version Name       SH: 'dcm4che-1.4.31'
```

```
.
```

```
.
```

```
.
```

اطلاعات بیشتر از این بسته را می‌توانید در [صفحه‌ی اصلی آن](#) بیابید.

^۱ فرق بیت‌های اختصاص داده با بیت‌های ذخیره شده این است که مثلاً برای تصاویر ۶ بیتی، ساختار آرایه‌ای مختص آن در نظر گرفته نشده، پس آن را درون یک ساختار ۸ بیتی ذخیره می‌کنند. ۸ تعداد بیت‌های اختصاص داده شده است و ۶ تعداد بیت‌های ذخیره شده است.

توضیحاتی در مورد کران نمایش تصویر

کران یک تصویر خاکستری، حد فاصل بین مقدار منسوب به سیاه و مقدار منسوب به سفید است. برای مثال در یک تصویر ۸ بیتی مقدار ۰ معادل سیاه و مقدار ۲۵۵ معادل سفید است. اعداد بین این‌ها به شکل خطی در طیف خاکستری قرار می‌گیرند. پس بنابراین کران یک تصویر، لزوماً بیشترین و کمترین مقدار آرایه‌ی تصویر نیست؛ برای مثال مقادیر یک تصویر ۸ بیتی می‌تواند بین ۲۰۰ تا ۲۴۰ باشد، اما همچنان کران آن $[0, 255]$ است. اگر ۲۰۰ را معادل سیاه و ۲۴۰ را معادل سفید در نظر بگیرید و نمایش بدهید، میزان روشنایی خالص تصویر را به‌درستی نمایش نداده‌اید و فقط میزان روشنایی نسبی تصویر را به‌درستی نمایش داده‌اید. رعایت کران نمایش تصویر در مقایسه‌ی روشنایی بین تصاویر اهمیت پیدا می‌کند. در `plt.imshow` اگر `vmin`, `vmax` را مشخص نکنید، خودکار کمینه و بیشینه‌ی مقادیر آرایه‌ی تصویر را به‌عنوان سیاه و سفید در نظر می‌گیرد. برای نمایش صحیح تصویر، محدوده‌ی ممکن مقادیر تصویر را به `vmin`, `vmax` مقداردهی کنید.

لینک‌های مفید برای حل مسائل ۲ و ۴

[Geometric Image Transformations](#)

[Geometric Transformations of Images](#)

۲ ۳۵%

(آ) از فایل file1.dcm تصویر را بخوانید. (۵%)

(ب) تعداد بیت‌های اختصاص داده شده (برای هر پیکسل)، تعداد بیت‌های ذخیره شده (برای هر پیکسل) و دسته‌بندی روش تصویربرداری و بخشی از بدن که از آن تصویربرداری شده است را استخراج و در خروجی چاپ کنید. (۱۰%)

(ج) با نمونه‌کاهی (downsample) ساده تعداد پیکسل‌های تصویر را در جهت محور x به یک‌چهارم کاهش دهید و نمایش دهید. همین کار را نیز جداگانه در جهت محور y تکرار کنید. (۱۰%)

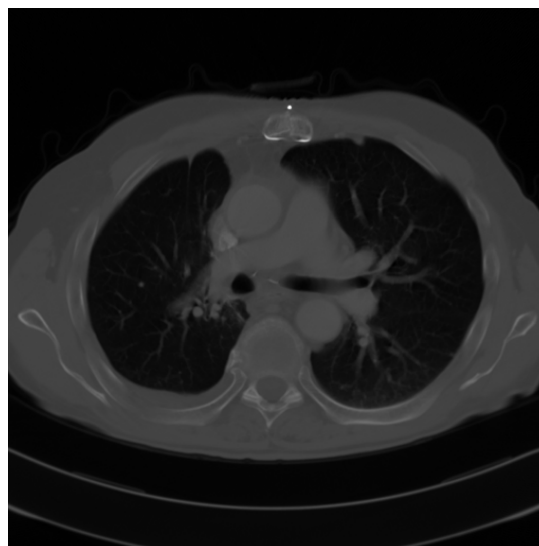
(د) با نمونه‌کاهی تعداد پیکسل‌های تصویر را در هر دو جهت نصف کنید و نمایش دهید. (۵%)

(ه) تصاویر بدست آمده از دو بخش قبلی را از لحاظ میزان صدمه دیدن نسبت به تصویر اصلی تحلیل و مقایسه کنید. می‌توانید از دیدگاه‌های متفاوتی به این موضوع نگاه کنید. (۱۰%)

(و) به کمک OpenCV و با روش‌های نزدیک‌ترین همسایه، دوخطی و دومکعبی تصویر بخش پیش را به ابعاد اولیه بازگردانید، نمایش دهید و در مورد تفاوت نتیجه‌ی هرکدام بحث کنید. (۲۰%)

(ز) تصویر را به شکل تصاویری با هشت، پنج، سه، دو و یک بیت (با نوع داده‌ی عدد صحیح) درآورید و با رعایت مقادیر vmin, vmax آن را با plt نمایش دهید. در مورد اثر کاهش عمق بیتی بر کیفیت تصویر بحث کنید. **پیاده‌سازی الگوریتم کاهش عمق بیتی جزو هدف مسأله است.** (۳۵%)

(ح) تصویر هشت‌بیتی شده را به فرمت‌های tif و bmp ذخیره کنید. (۵%)



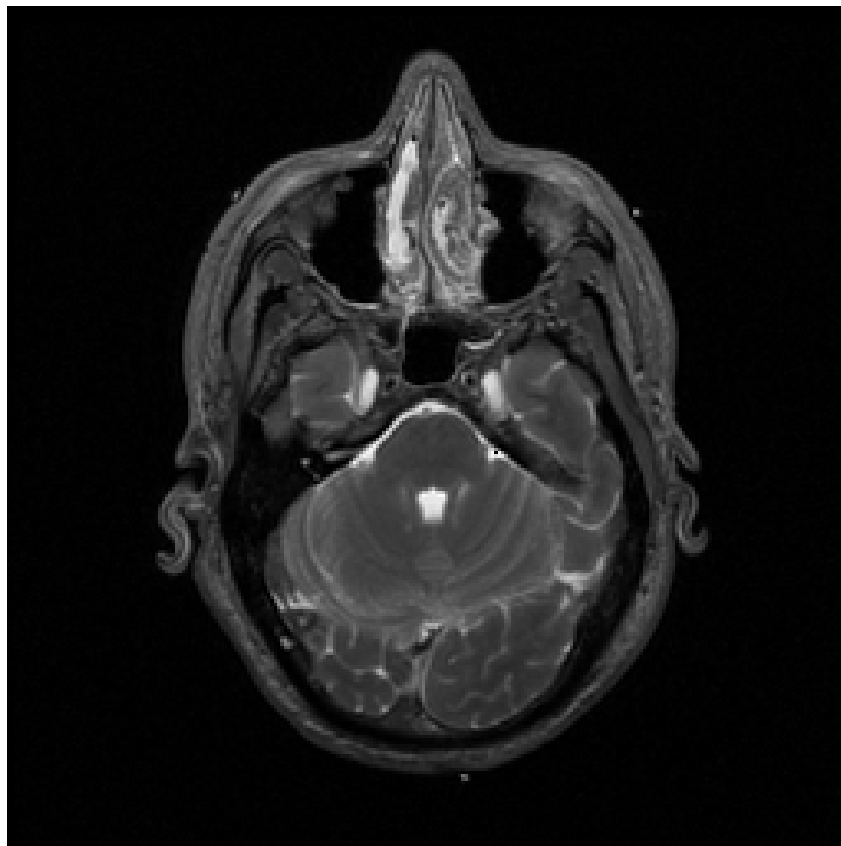
شکل ۱: [۱]

۳ ۱۵%

آ) فایل ویدیویی MRI-Head.avi را بخوانید. ویدیو شامل تصاویری نویزی (با توزیع نرمال) از یک صحنه‌اند. از تمام فریم‌های آن میانگین بگیرید تا نویز تصویر کاهش پیدا کند. فریم اول ویدیو و تصویر میانگین گرفته شده را در کنار یکدیگر نمایش دهید تا اثر کاهش نویز نمایان شود. (۶۰٪)

ب) این کار را با استفاده از نصفه‌ی اول فریم‌ها^۲ تکرار کنید و تفاوت بین دو خروجی و علت آن را بیان کنید. (۱۵٪)

ج) با تابع `np.load()` دو فایل `mask1.npy` و `mask2.npy` را بخوانید، هرکدام از آرایه‌های داخل این فایل‌ها را جداگانه در تصویر میانگین گرفته شده ضرب کنید، نتایج را با یکدیگر جمع کنید و نمایش دهید. عملیات رخ داده را شرح دهید. (۲۵٪)



^۲ یعنی اگر $2n$ فریم در فایل ویدیویی وجود دارد، با n فریم اول این کار را انجام دهید.

۴ ۲۵%

در هنگام کدنویسی، به تفاوت تعاریف محورها بین OpenCV و تئوری درس توجه کنید.

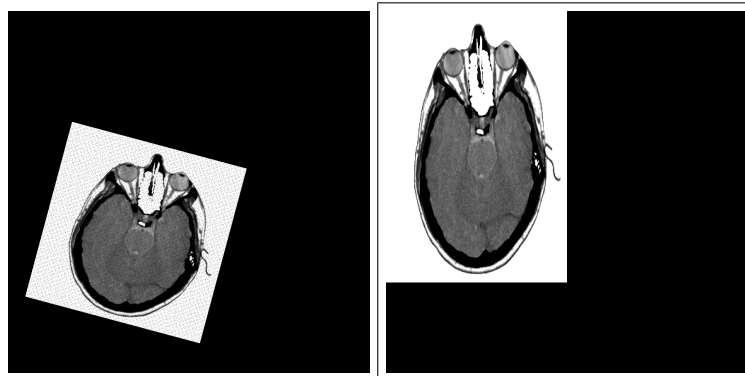
آ) تصویر transformed.png را بخوانید. ماتریس تبدیل هندسی‌ای که روی آن اعمال شده است را حدس بزنید. تصویر اولیه تمام آرایه‌ی تصویر را پوشانده بود و هیچ ناحیه‌ی مستطیلی سیاهی نداشت. نوع تبدیل را در گزارش بیان کنید. (۲۰%)

ب) تصویر را با اعمال تبدیل هندسی مناسب به حالت اصلی خود برگردانید و بعد در چهار طرف تصویر یکسان به‌میزانی مقدار صفر لایه‌گذاری (padding) کنید که ابعاد تصویر دوبار برابر شود. در بخش‌های بعدی منظور از «تصویر» خروجی این بخش است. (۲۰%)

ج) تصویر را با استفاده از تبدیل جابجایی (Translation) به میزان ۱۰۰ پیکسل به بالا و ۲۵۰ پیکسل به راست جابجا کنید. (۲۰%)

د) تبدیل shear را با $s_v = 0, s_h = -0.3$ به تصویر اعمال کنید. (۲۰%)

ه) تصویر را از روش backward به اندازه‌ی ۱۵ درجه نسبت به مبدأ مختصات و ساعتگرد بچرخانید. نتیجه‌ی روش forward در فایل forward_image.jpg وجود دارد. در مورد تفاوت خروجی این دور روش بحث کنید. (۲۰%)



نحوه‌ی ارسال: فایل PDF گزارش به‌همراه کدهای نوشته‌شده (py) در قالب یک فایل فشرده‌ی zip به اسم HW1_Num باشند که Num شماره‌ی دانشجویی شما است (مانند HW1_400133001) و فقط از طریق سامانه‌ی مدیریت یادگیری **Courses** ارسال بفرمایید. موفق باشید.

مراجع

- [1] Albertina, B., Watson M. Holback C. Jarosz R. Kirk S. Lee Y. ... Lemmerman J. Radiology Data from The Cancer Genome Atlas Lung Adenocarcinoma [TCGA-LUAD] collection, 2016.