به نام خدا

گزارش تمرین شماره یک پردازش تصویر

دکتر آذرنوش

امیرحسین شریفی صدر 9733044

** به دلیل نوشتن برنامه ها با jupyter notebook برای خواندن تصاویر از آدرس آن ها در کامپیوتر خودم استفاده شده که ممکن است برای اجرا در سیستم های دیگر کد ها اجرا نشوند . اما تمام خروجی ها یا در گزارش یا در فایل تحویلی قابل مشاهده هستند و این مورد در نظر گرفته شود . ممنون .

سوال شماره 1 :

در بخش الف سوال شماره یک از ما ابعاد و داده هر پیکسل خواسته شده که بخشی از خروجی را در این قسمت به نمایش میگذارم :

```
(512, 512, 3)
[[[ 89 143 160]
  [  7  59  75]
  [  0  47  61]
  ...
  [ 74 119 116]
  [115 161 155]
  [137 185 179]]

 [[ 66 118 135]
  [ 46  98 114]
  [  0  43  55]
  ...
  [ 96 136 131]
  [108 152 146]
  [ 85 129 122]]

 [[ 24  75  91]
  [ 70 122 135]
  [  0  49  59]
  ...
  [ 60  91  88]
  [ 66 102  96]
  [ 51  90  82]]

 ...
```

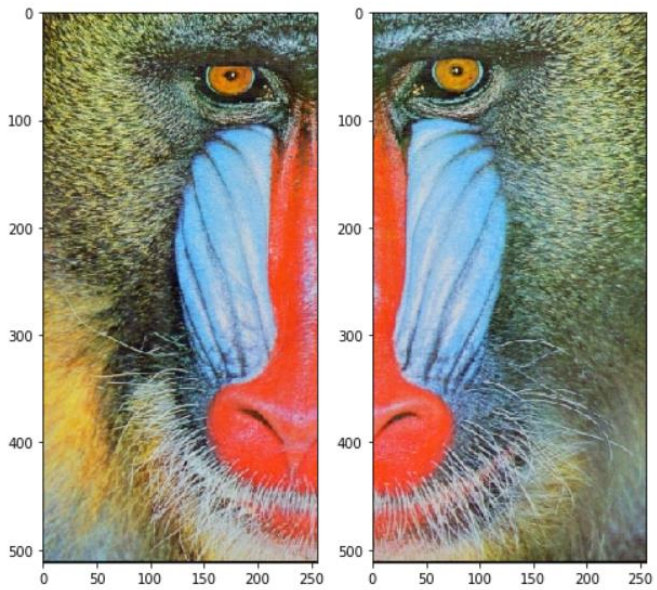
در بخش های دیگر سوال از ما خواسته تا اعمالی را روی تصویر انجام دهیم که آنها در فایل تحویلی به صورت cv.imwrite() سیو شده اند :

بخش ب : img_gray.jpg

بخش ج : img64.jpg , img16.jpg , img2.jpg

که در بخش ج میبینیم که به دلیل تغییر دادن سطح روشنایی از 255 به 64 و16 و 2 تصاویر بسیار تاریک و غیر قابل مشهاده میشوند چون بازه رنگ ها بسیار کم میشود برای مثلا در سطح روشنایی 2 تنها مجموعه ای از سه مقدار 0 , 1 , 2 برای هر پیکسل به عنوان رنگ میتواند نمایش داده شود که عملا تصویر سیاه میشود .

بخش د : هم به صورت plot در خروجی به نمایش گذاشته شده و هم در فایل تحویلی دو تصویر کراپ شده به تفکیک قابل مشاهده هستند . left of crop.jpg , right of crop.jpg



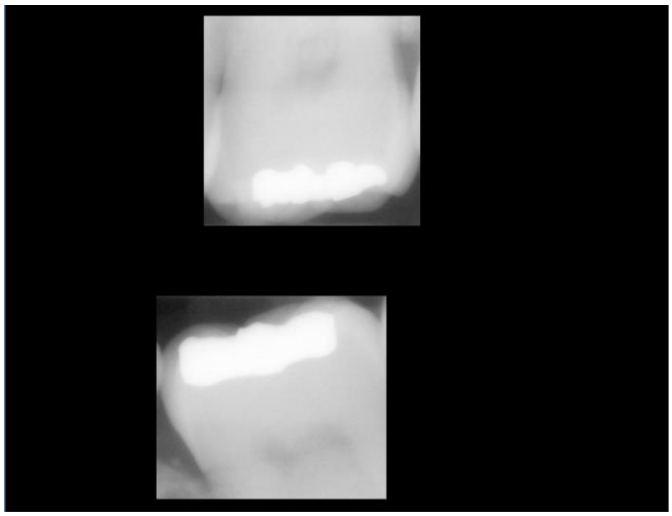
بخش ه : bottom to top inverse.jpg , right to left inverse.jpg

بخش و : gray flip.png

بخش ی : در کد بزرگنمایی و کوچکنمایی به سه روش با متد های cv.INTERNEAREST , cv.INTERLINEAR , cv.INTERAREA انجام شده است که در فایل تحویلی بزرگنمایی و کوچکنمایی با روش cv. INTERLINEAR قابل مشاهده است

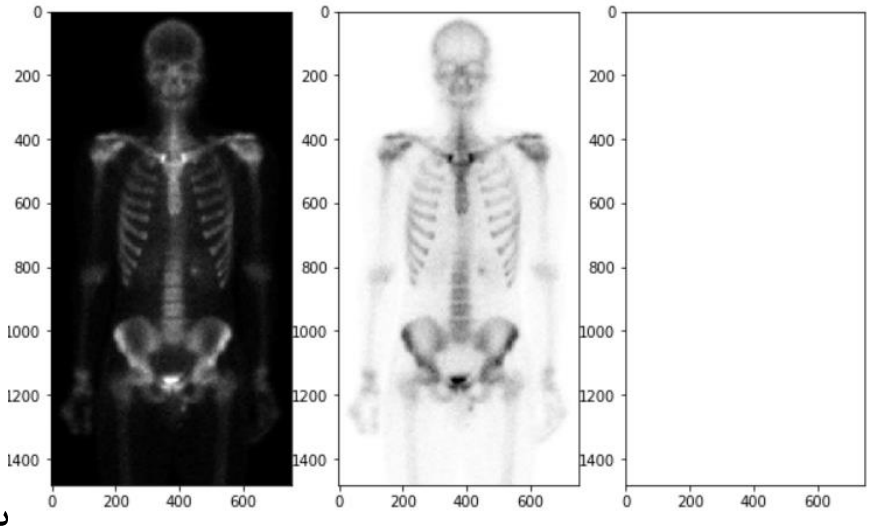
img_withdraw1.jpg , img_stretch1.jpg

سوال شماره 2 : در بخش الف خواسته شده که بخش های مشخص شده در تصویر ماسک را از تصویر اصلی استخراج کنیم که در زیر قابل مشاهده است :

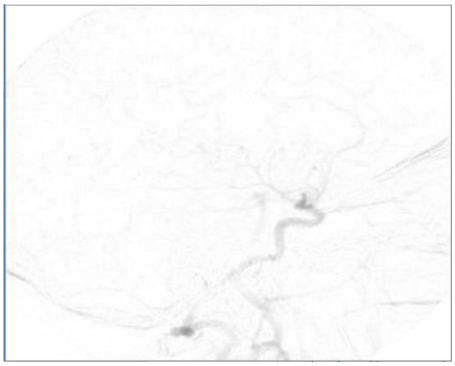


در بخش ب ابتدا تصویر را به صورت خاکستری میخوانیم سپس مکمل آن را با کم کردن داریه های هر پیکسل از 255 به دست میاوریم و در

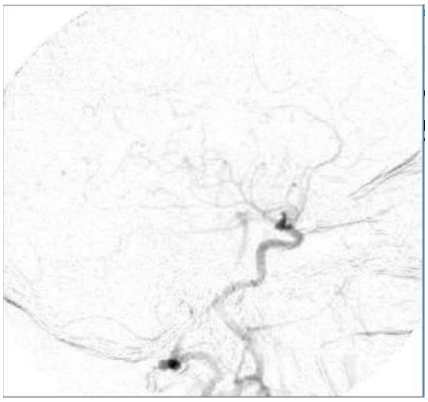
آخر اجتماع تصویر اول و تصویر مکمکل را به دست میاوریم که تصویری تماما سفید است که البته با توجه به روشی اجتماع گیری استفاده شده در کتاب پردازش تصویر دیجیتال برای همین عکس نتیجه چیز دیگریست که متاسفانه بنده متوجه روش استفاده شده در کتاب نشدم و از تکنیک جمع کردن دو عکس استفاده نمودم که چون مکمل یکدیگر هستند حمع مقادیر هر پیکسل 255 میشود و تصویر سفید .



بخش ج : با استفاده از تابع cv.absdiff تفاوت پیکسل های بین دو تصویر live و maskرا به دست آورده :



و حال با نرمالایز کردن پیکسل ها یعنی قرار دادن بازه مقدار در هر پیکسل بین 0 تا 255 تصویر نهایی و بهتری خواهیم داشت :

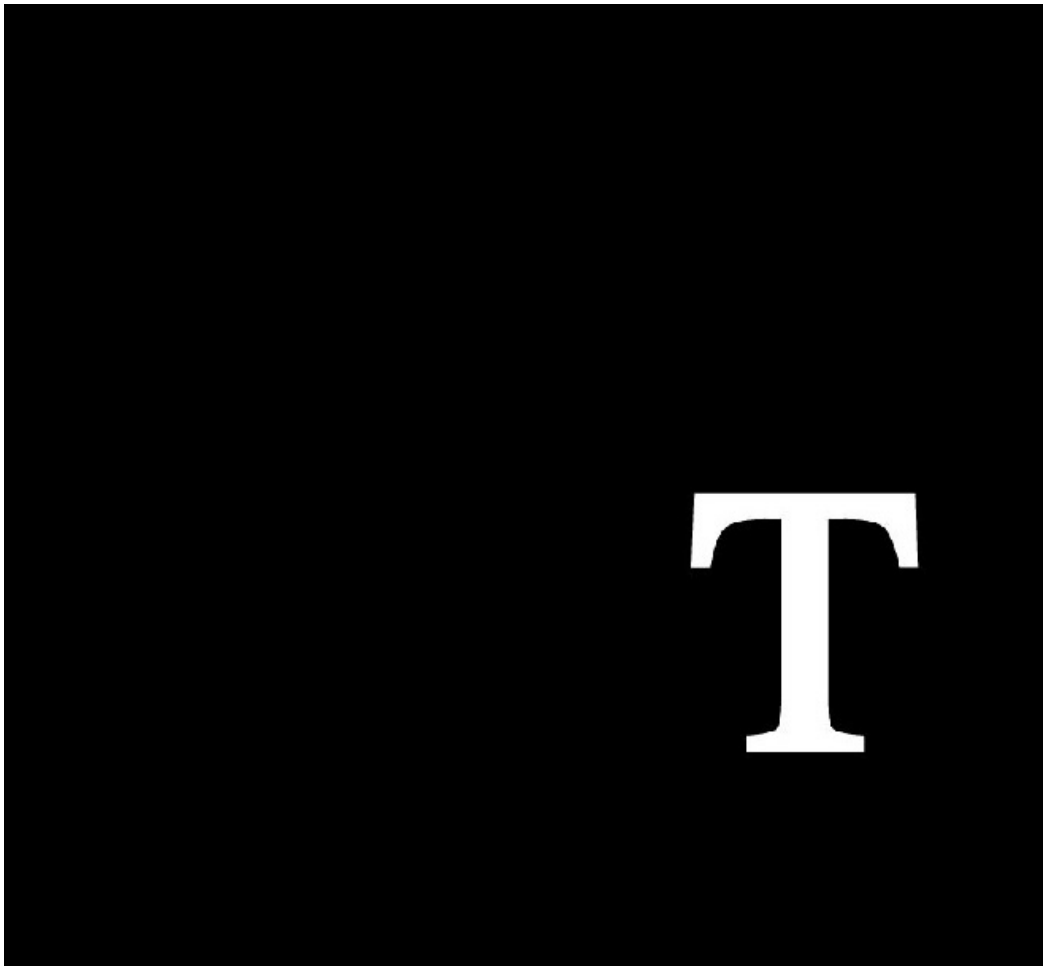


سوال شماره 3 :

در بخش اول سوال عملیات scale در جهت کوچک کردن تصویر با پارامتر $\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

در بخش دوم سوال عملایت translation با پارامتر :

$$\begin{bmatrix} 1 & 0 & 200 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix}$$

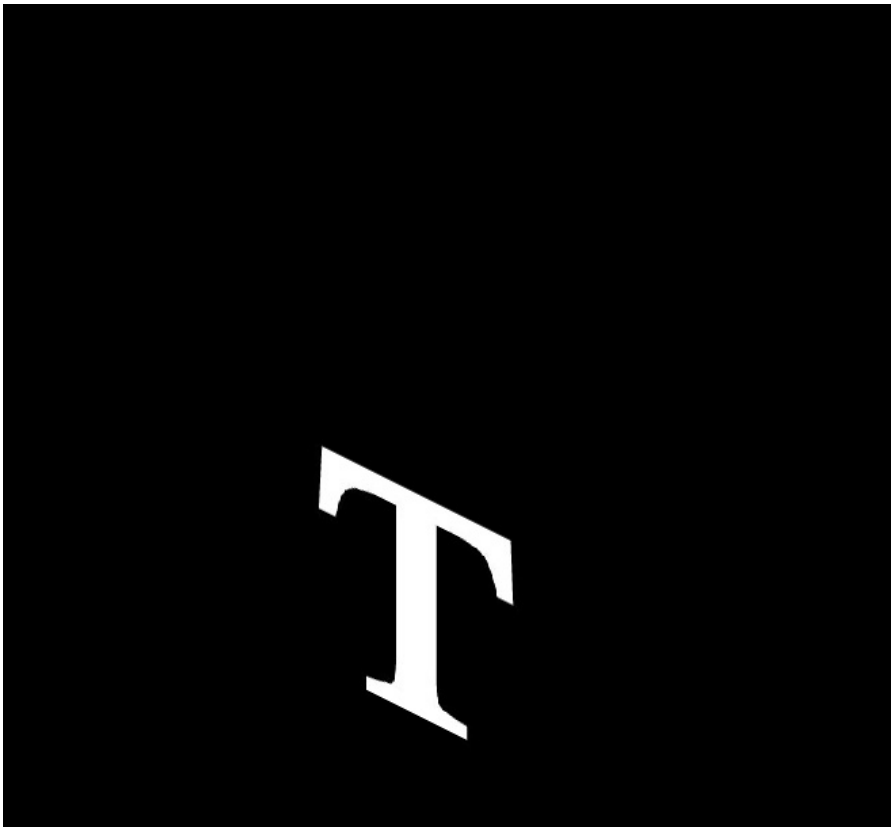


در بخش سوم سوال عملیات vertical shear با پارامتر :

$$\begin{bmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

در بخش چهارم سوال عملیات horizental shear با پارامتر :

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



و به طور کلی برای عملیات scale از تابع cv.resize() و برای سه عملیات دیگر از تابع cv.warpAffine استفاده شده است .

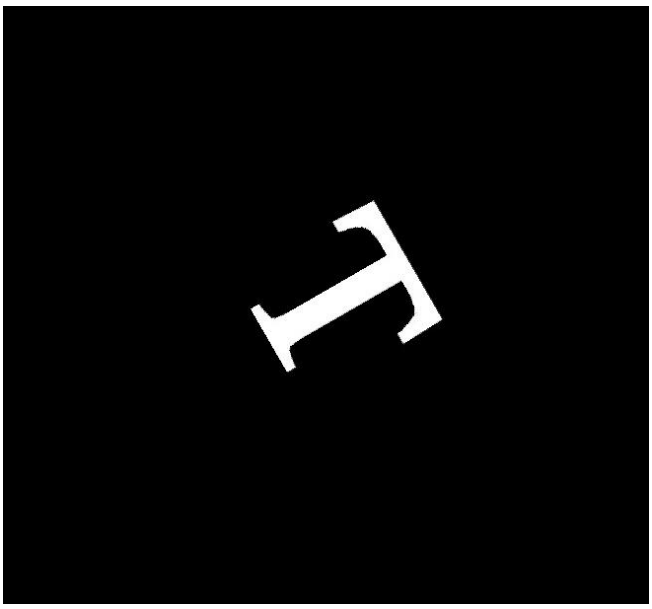
برای بخش آخر سوال که عملیات rotation خواسته شده از دو روش forward mapping و backward mapping استفاده شده است .

Forward mapping به این شکل است که روی هر پیکسل ورودی تکرار میشود و مقدار و مختصاتی جدید برای آن به وجود می آورد و آن را روی پیکسل تصویر خروجی کپی میکند که از مشکل های این روش میتوان به این اشاره کرد مختصات جدید به دست آمده برای هر پیکسل ممکن است در بازه طول و عرض تصویر نباشد که این مورد چک میشود و همچنین ممکن است دارای مقدار صحیح یا integer نباشد که این مشکل نیز با پیدا کردن نزدیک ترین عدد صحیح به آن و جایگزین کردنش حل میشود . مشکلی نیز وجود دارد که ممکن است چند پیکسل از ورودی بر از تغییر دارای مختصات یکسان شوند یا همان مشکل که در بالا عرض کردیم در بازه مورد نظر نباشند و نتوان آنها را در تصویر خروجی قرار داد که در این صورت تصویر خروجی دارای حفره هایی میشود .

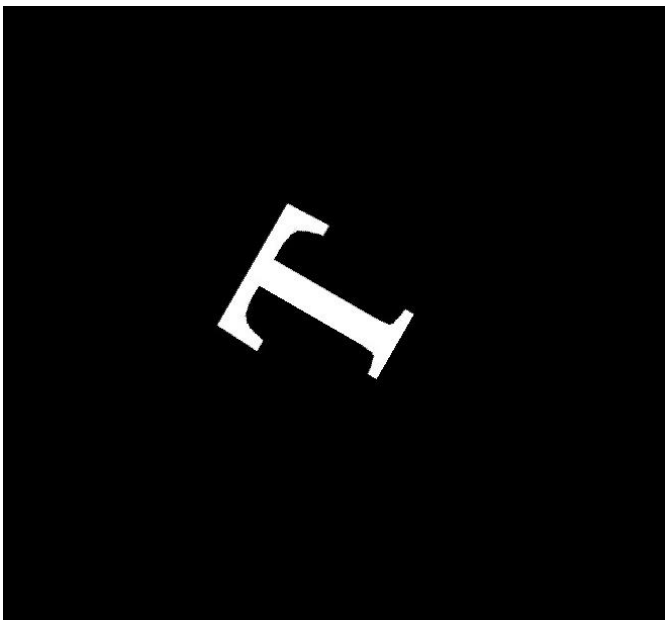
Backward mapping برای برگرداندن تصویر بعد از تغییر به حالت اولیه آن توسط عملیات inverse یا معکوس سازی است . در این روش نیز دو مشکل وجود داشته در forward mapping که همان صحیح نبودن و در بازه مشخص شده نبوده مختصات ها ی جدید است وجود دارد که قابل حل است ولی در این روش در تصویر نهایی ما حفره ای وجود نخواهد داشت .

بنده در برنامه از backward برای چرخاندن تصویر اصلی به سمت چپ استفاده کردم نه برای برگرداندن تصویر ایجاد شده توسط forward به حالت اولیه .

Forward Mapping



Backward Mapping



سوال شماره 4 : تصویر نمایش داده شده threshold شده است .