به نام خدا

پردازش تصویر

تمرین شماره ۱

آشنایی با توابع کتابخانه ای OpenCV

امین سخایی

۹۷۳۳۰۳۶

استاد درس

دکتر حامد آذرنوش

سوال شماره ۱:

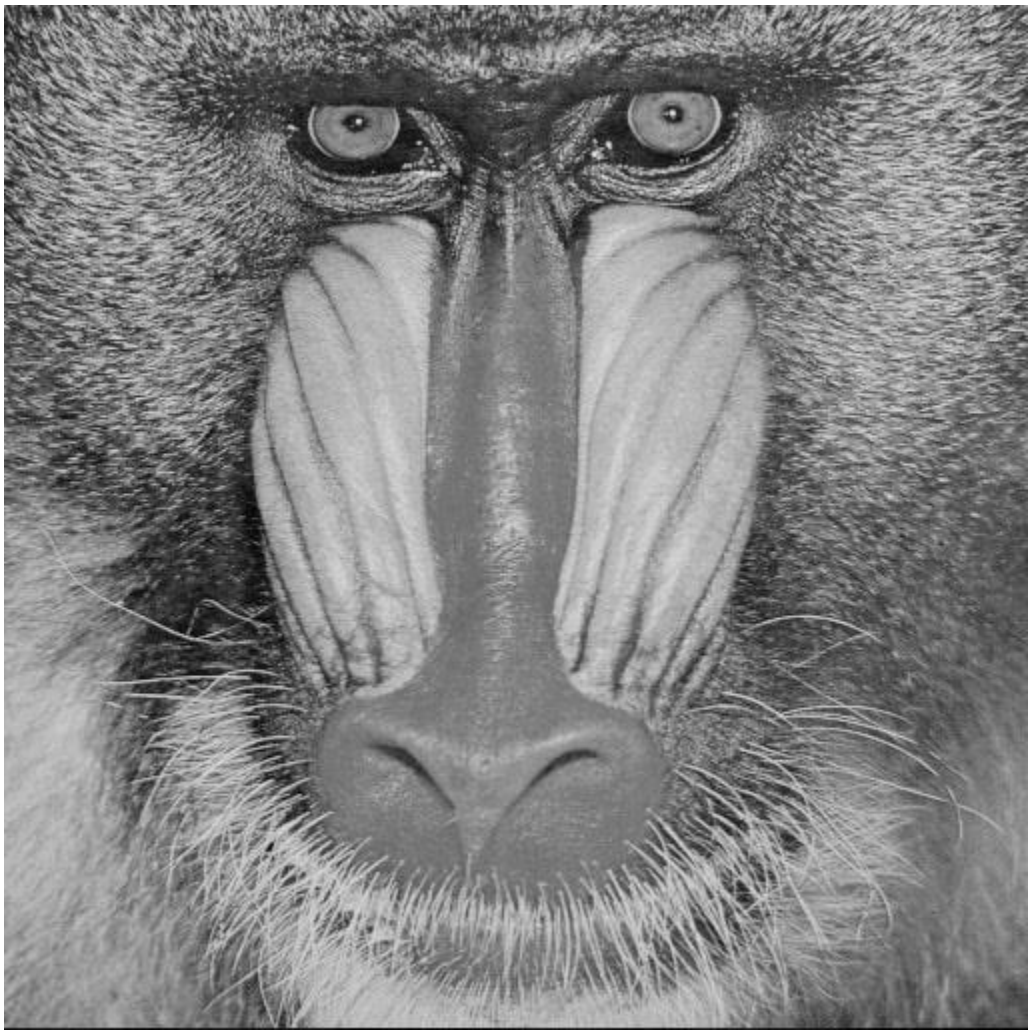
وردی برنامه تصویر mandrill.jpg است که استفاده از توابع آماده آنرا میخوانیم.

بخش الف) نوع داده تصویر یک آرایه استاندارد است که توسط کتابخانه ی numpy تعریف می شود و شامل پیکسل های متناظر با نقاط داده ای است.

خروجی:

```
(512, 512, 3)
<class 'numpy.ndarray'>
```

بخش ب) خروجی:



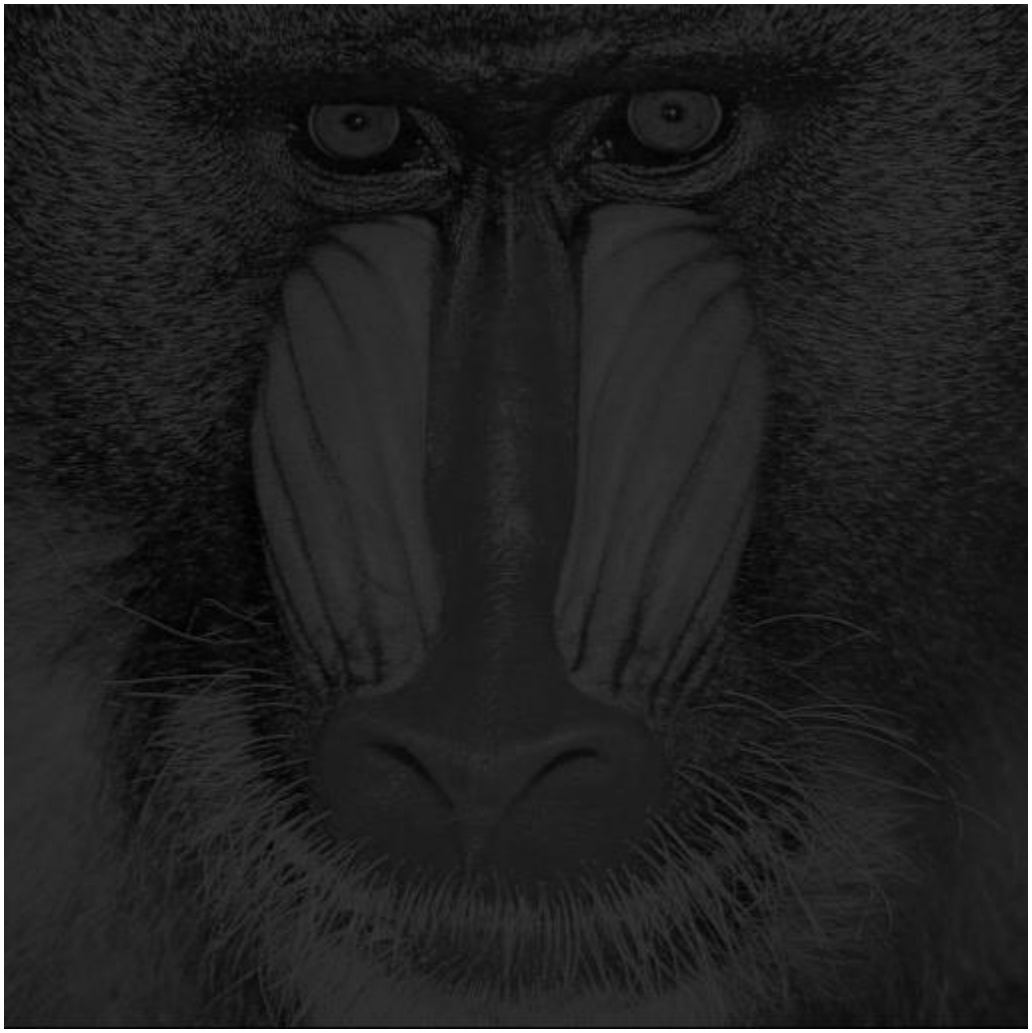
بخش ج)برای تغییر فضای رنگی از [۰, ۲۵۵] به [۰,۷] از فرمول زیر استفاده میکنیم:

$$x(new) = [\frac{(x(old) - ۰)}{(۲۵۵ - ۰)} * y]$$

تصویر ورودی یک تصویر grayscale با ۲۵۶ سطح روشنایی است ، بنابراین هریک از پیکسل های تصویر در شدت روشنایی در بازه ی [۰, ۲۵۵] هستند. با کم کردن سطح روشنایی، بازه ی شدت روشنایی پیکسل ها در محدوده ی کوچک تری قرار میگیرد و باعث می شود که نویز های تصویر نمایش داده نشود همچنین در سطوح روشنایی خیلی پایین جزییات تصویر ازبین می رود و تشخیص تصویر دچار مشکل می شود.

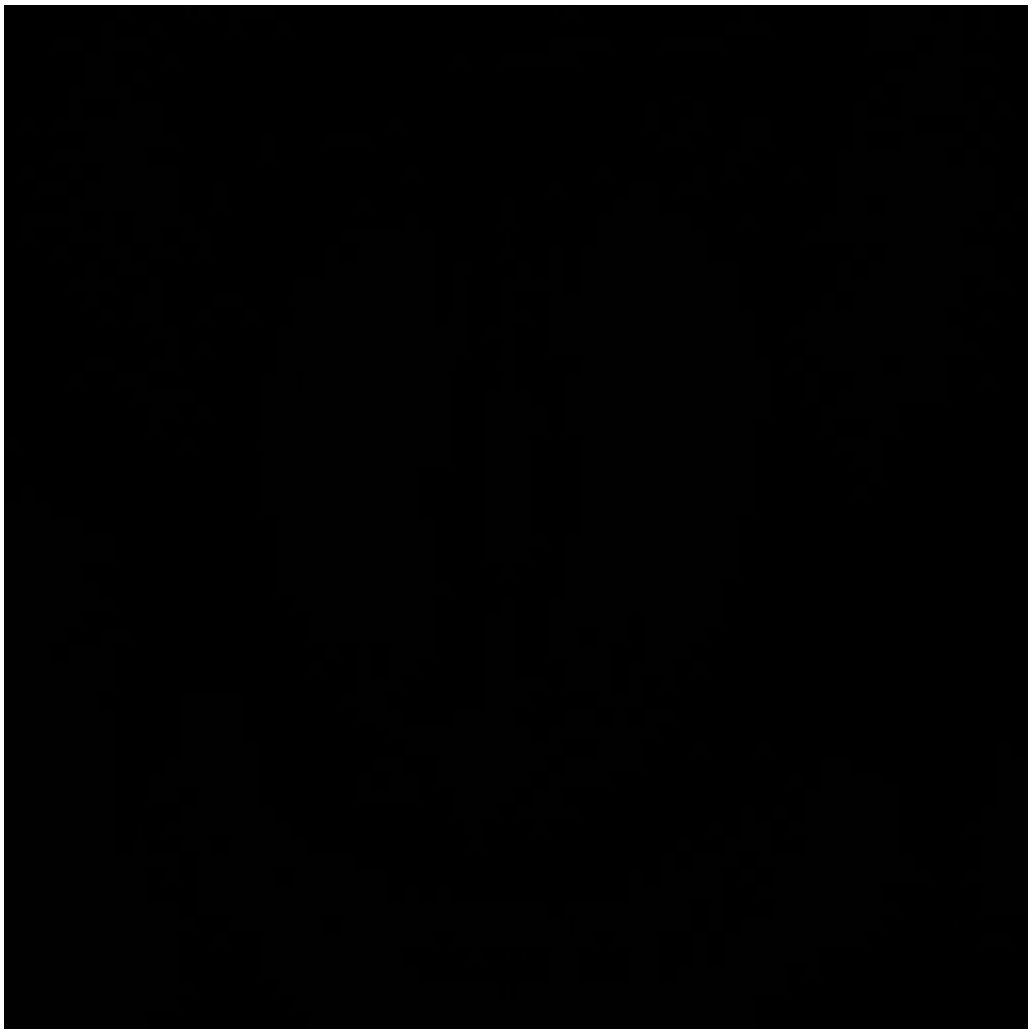
خروجی:

سطح روشنایی ۶۴



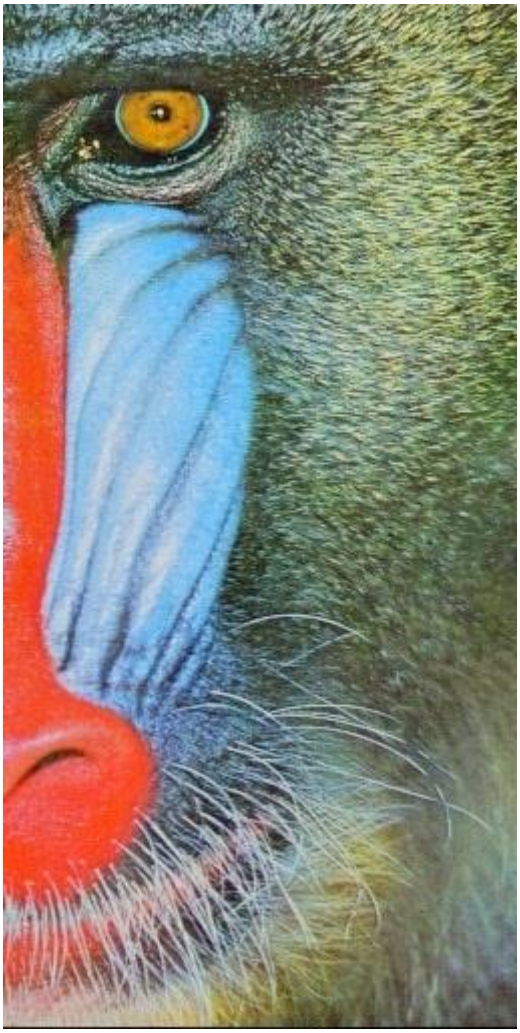
سطح روشنایی ۱۶

سطح روشنایی ۲



بخش د) برای انجام کراپ از دو ماتریس استفاده میکنیم که مقادیر سطر های آنها برابر با مقادیر سطرهای تصویر اصلی است و از مقادیر ستون های [۰, ۲۵۶] را در ماتریس اول و مقادیر ستون های[۵۱۲ ,۲۵۶] را ماتریس دوم کپی می کنیم.

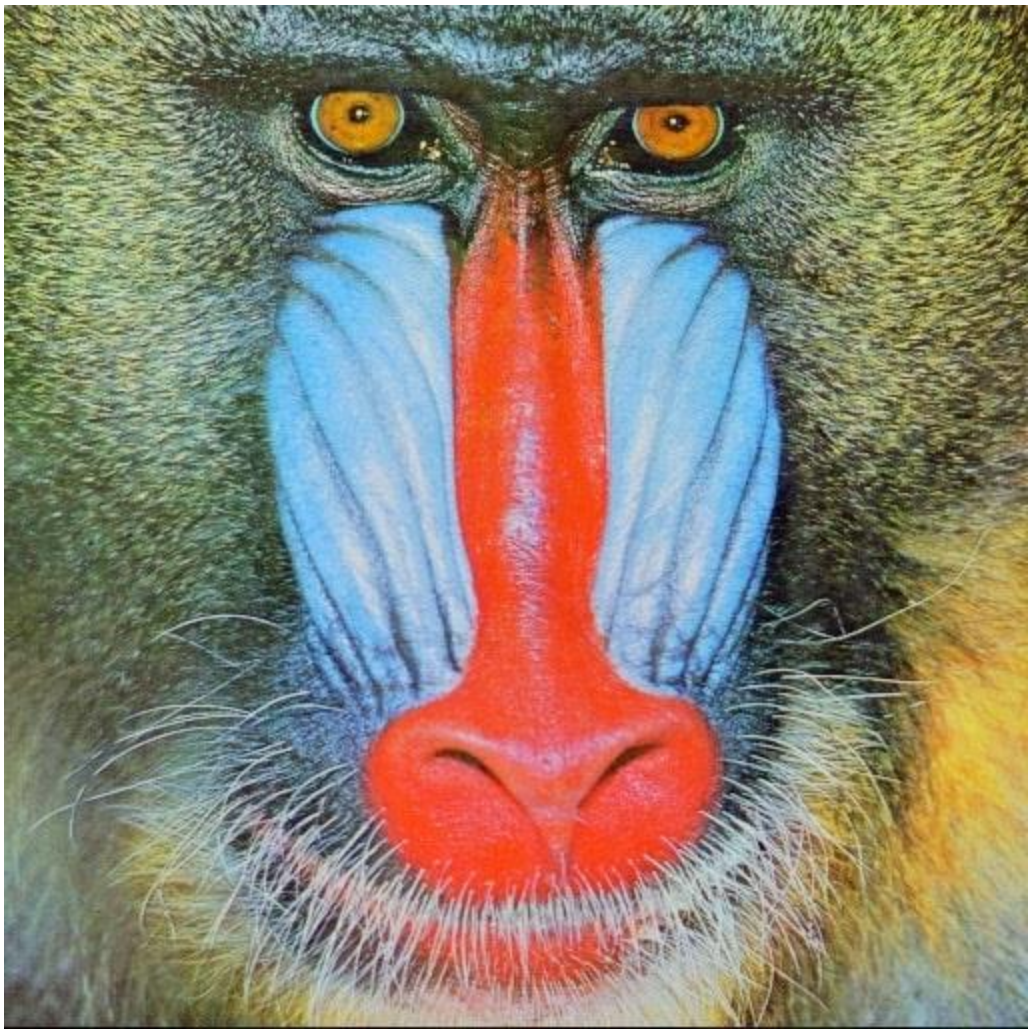
خروجی:



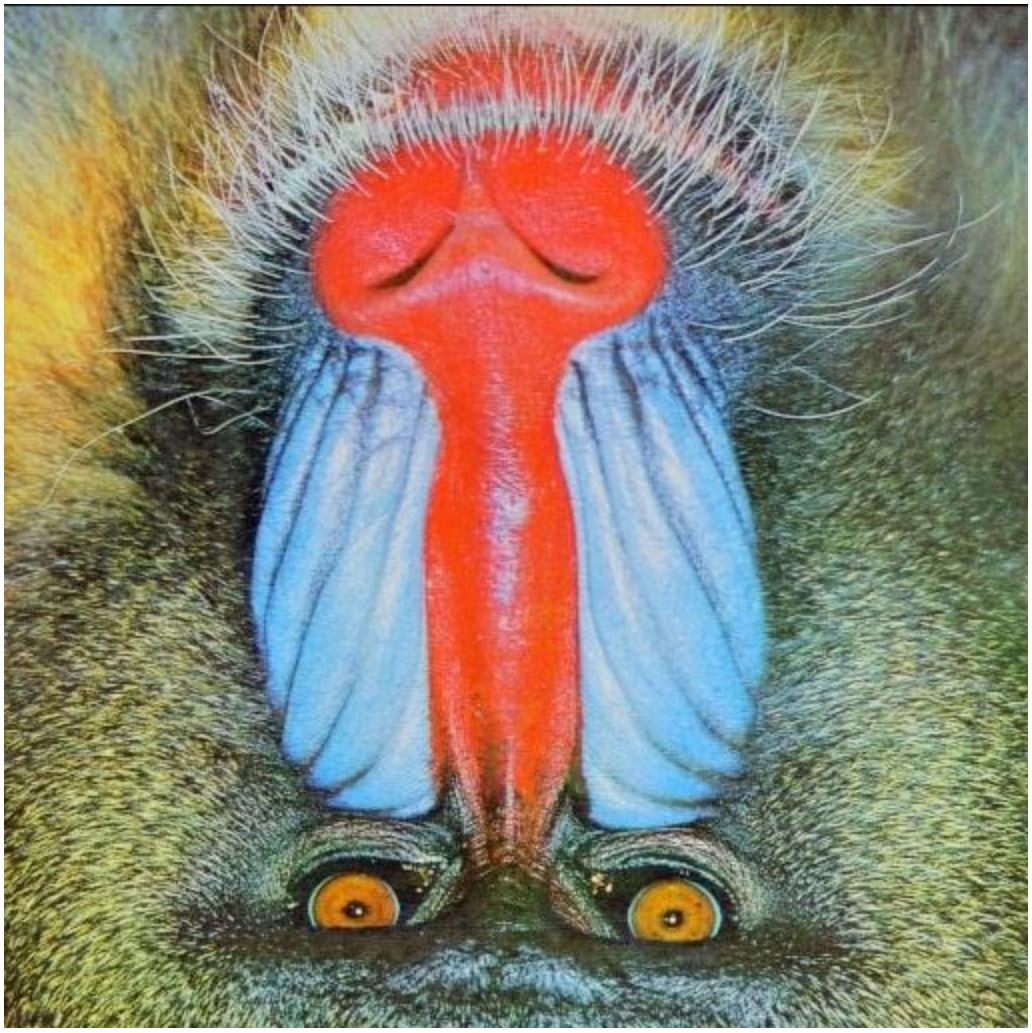
بخش ه)

خروجی:

وارون شده از راست به چپ



وارون شده از بالا به پایین



بخش ی)اندازه تصویر و رزولوشن با هم رابطه ی عکس دارند. به طوریکه با افزایش سایز تصویر، رزولوشن کاهش می یابد و بلعکس.برای از بین بردن افت کیفیت تصویر در هنگام تغییر ابعاد از درون یابی استفاده می کنیم.

درون یابی دوخطی:

این روش ارزش یک پیکسل رنگی را براساس چهار پیکسل در جهت‌های عمودی و افقی پیکسل در تصویر اصلی معین‌می‌کند. تصویر جدید دارای خاصیت Anti-aliasing است و تقریباً هیچ اثری از پیکسل‌های شطرنجی در آن دیده نمی‌شود. درون یابی دوخطی، تصویر بسیار هموارتری از درون یابی نزدیکترین همسایگی تولید می کند. اگر چه نیاز به زمان پردازش بیشتری نسبت به روش قبلی دارد، کیفیت نهایی تصویر خروجی به شدت بهتر می شود.

درون یابی نزدیک ترین همسایه:

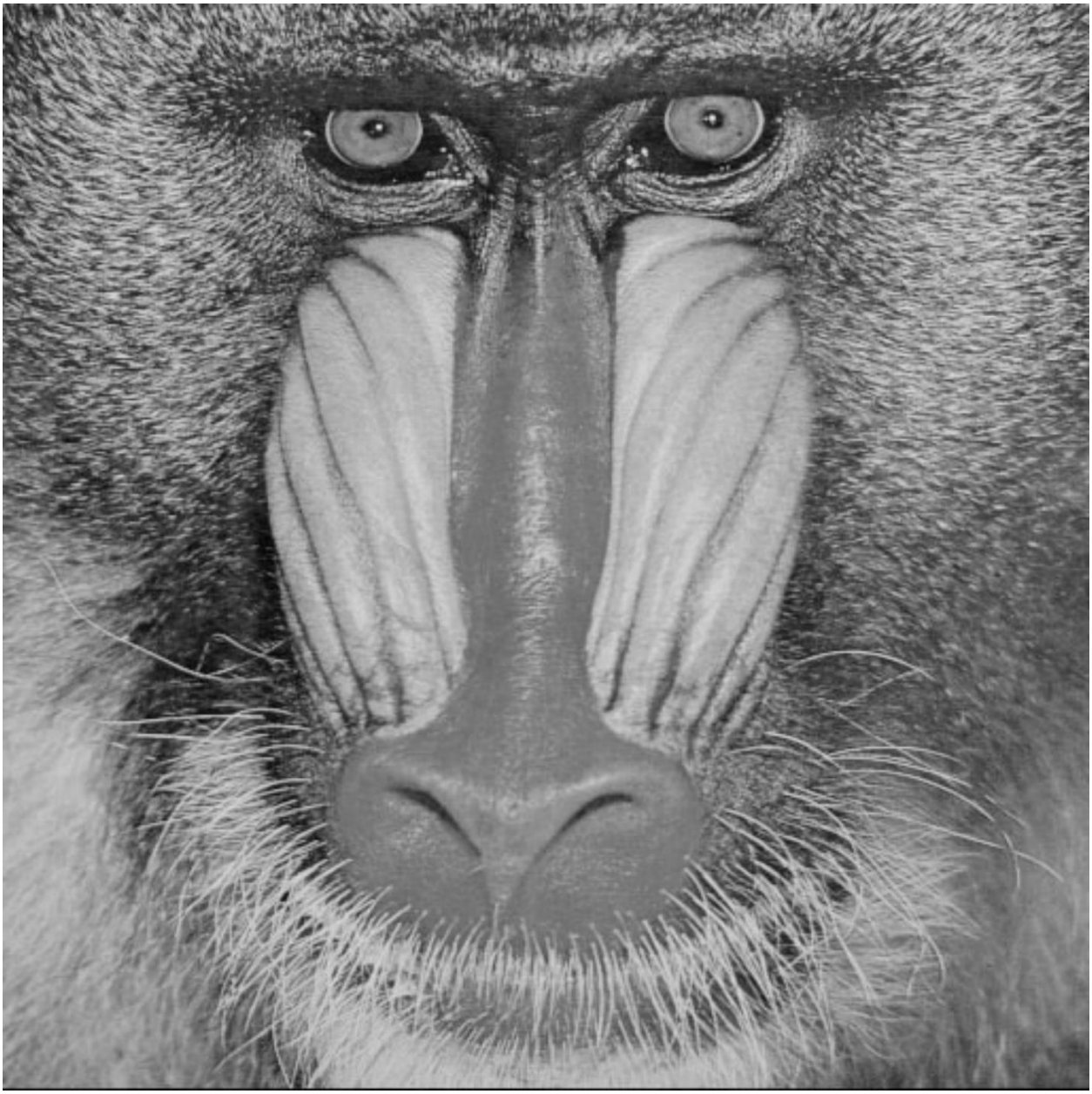
این روش ساده‌ترین عملیات درون‌یابی محسوب می‌شود که اساسا پیکسل‌های بزرگ‌تری را ایجاد می‌کند و رنگ هر پیکسل در تصویر جدید، با رنگ نزدیک ترین پیکسل در تصویر اصلی مطابقت دارد. اصولا بهتر است از این روش برای افزایش ابعاد تصویر استفاده نشود. چرا که موجب ایجاد حالت شطرنجی در تصویر می‌گردد.

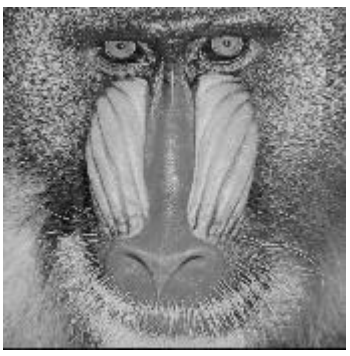
درون یابی تکرار پیکسل ها:

با استفاده از رابطه منطقه پیکسل ، نمونه برداری مجدد انجام می شود. تکرار پیکسل ها یک روش ارجح برای کاهش تصویر باشد ، زیرا نتایج بدون moir را به همراه دارد اما در هنگام بزرگنمایی مانند تصویر مانند نزدیک ترین همسایه عمل می کند.

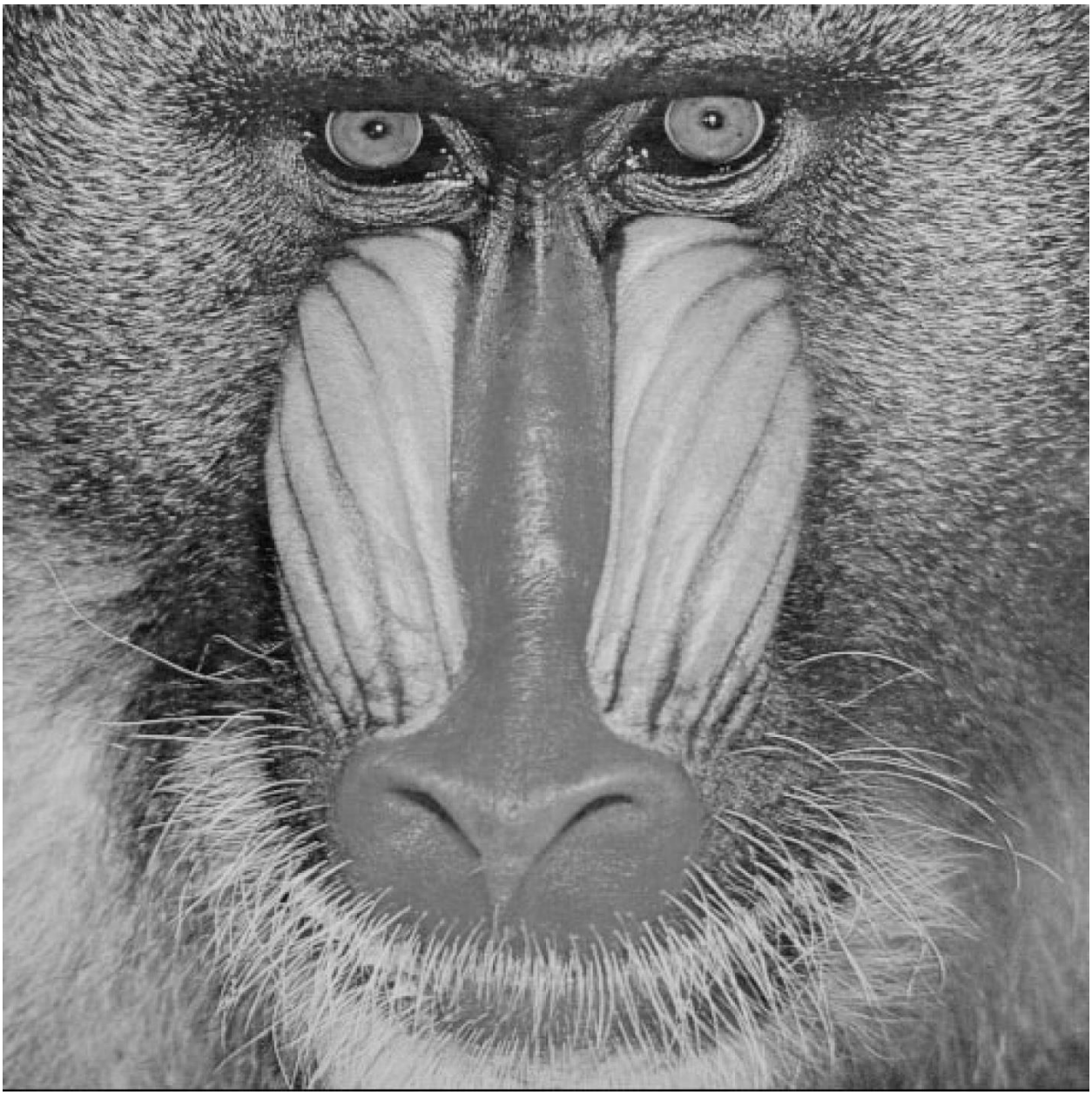
خروجی:

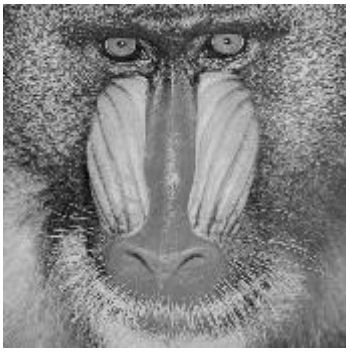
تغییر ابعاد بوسیله درون یابی دو خطی



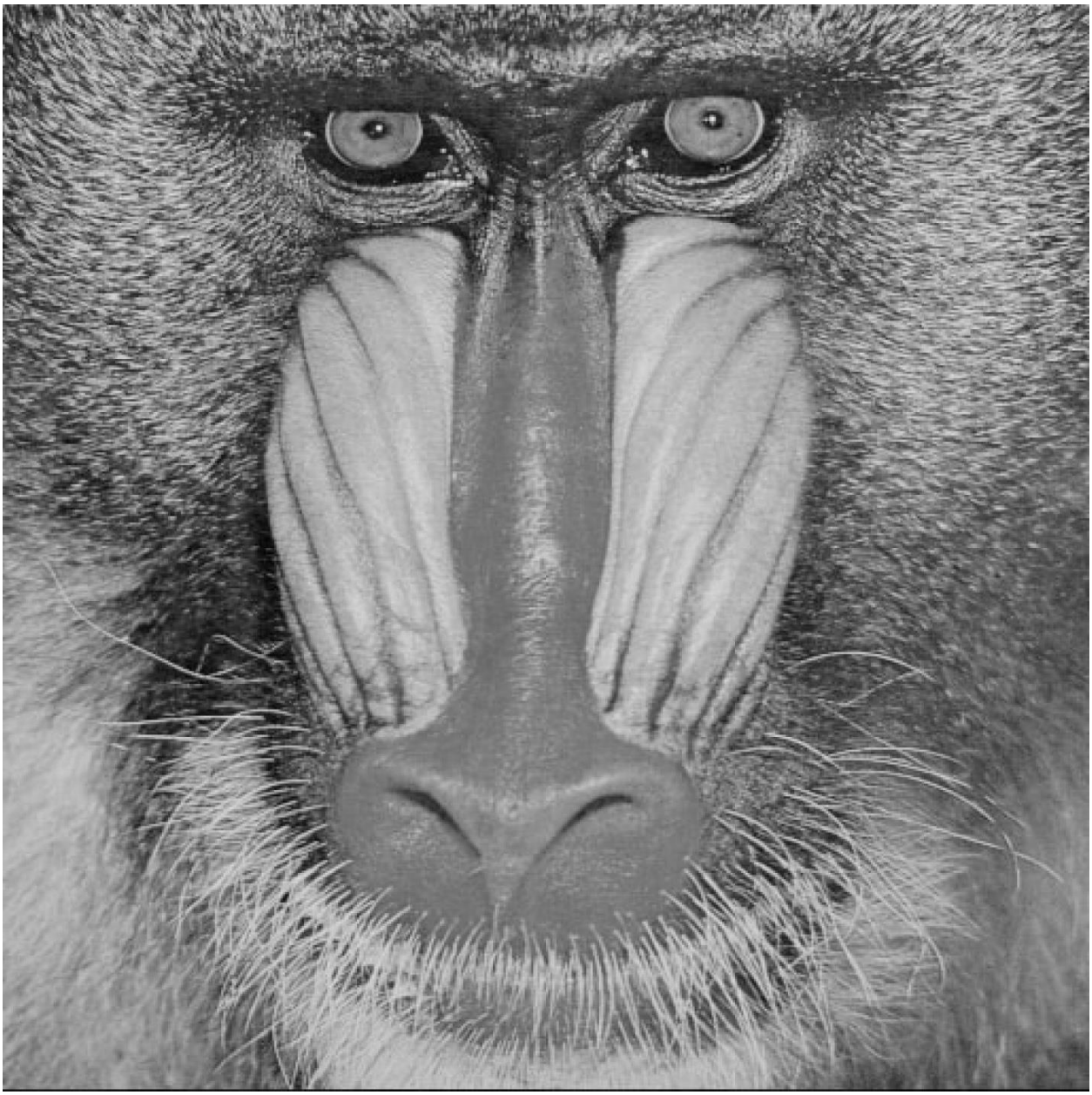


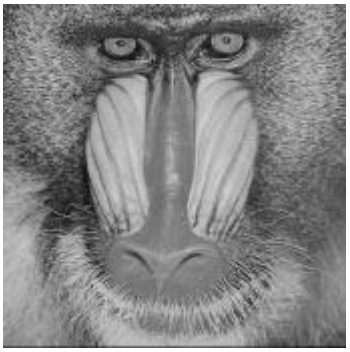
درون یابی نزدیک ترین همسایه





درون یابی بوسیله تکرار پیکسل ها



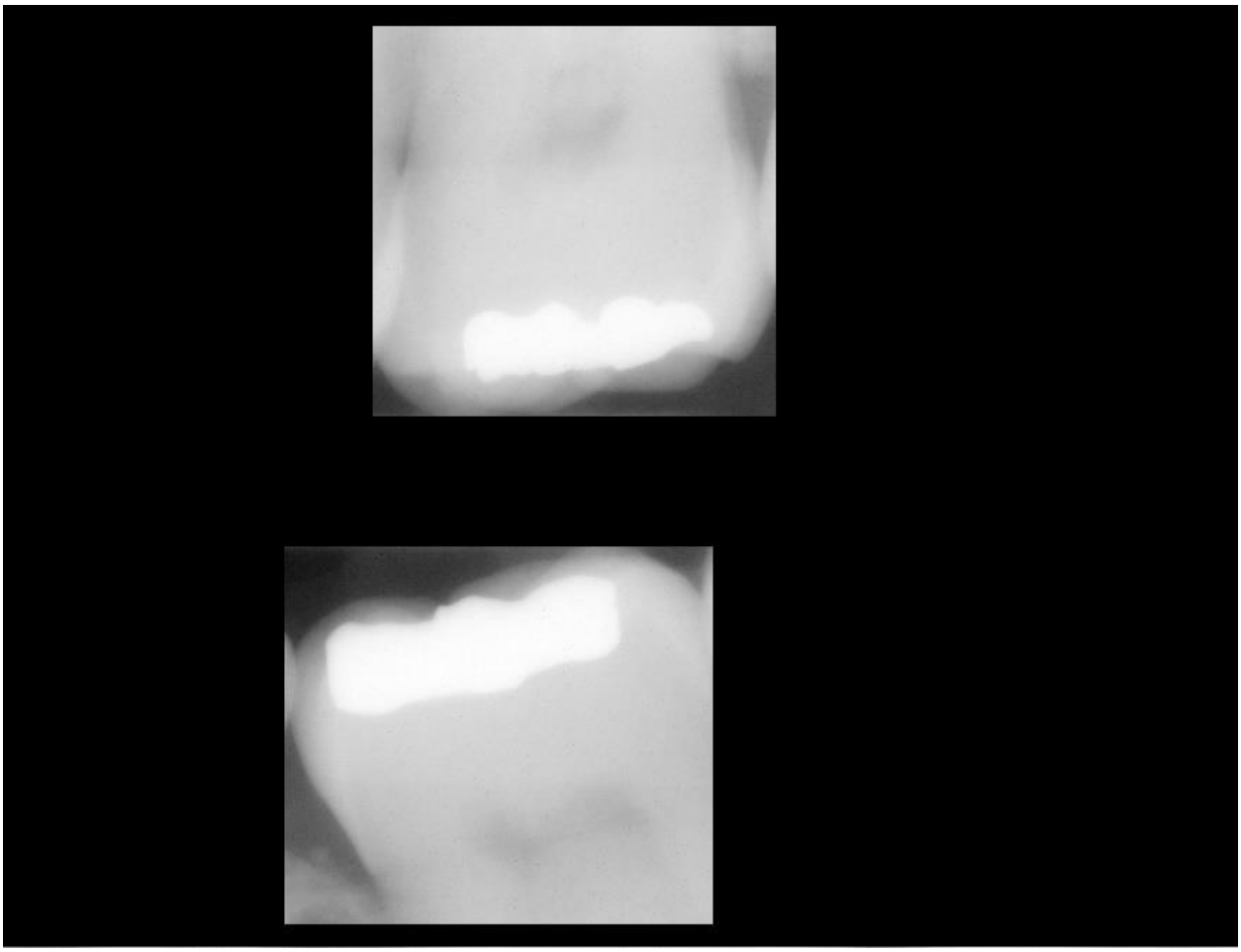


سوال شماره ۲:

بخش الف)

ورودی برنامه دو تصویر dental_xray_mask.tif وdental_xray.tif هستند. برای سهولت انجام عمیلیات منطقی بر روی تصاویر خوانده شده آن ها را به grayscale تبدیل میکنیم. سپس در یک حلقه ی تکرار در ماتریس تصویر اصلی و استفاده از دستور شرطی هرگاه در تصویر mask به مقدار صفر رسیدیم آرایه متناظر در تصویر اصلی را برابر با عدد صفر قرار می دهیم تا ناحیه مشخص شده را استخراج کنیم.

خروجی:

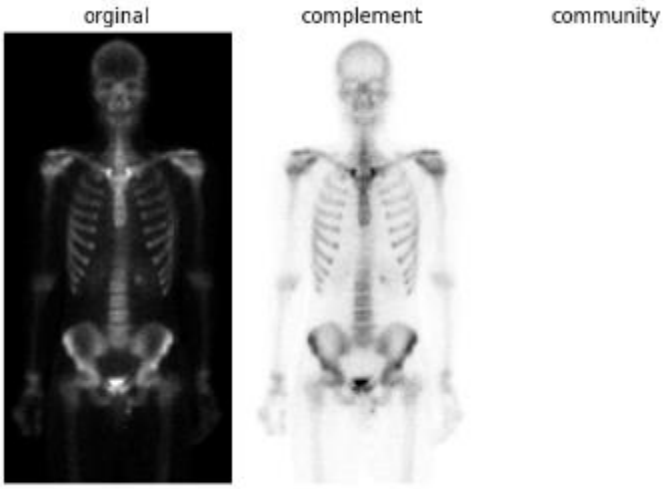


بخش ب)

ورودی برنامه تصویر partial_body_scan.tif که آنرا به صورت خاکستری می خوانیم. برای مکمل کردن تصویر باید پیکسل های با مقدار ۲۵۵ را به ۰ و پیکسل های با مقدار ۰ را به ۲۵۵ برسانیم.برای انجام این کار باید مقادیر ماتریس را از عدد ۲۵۵ کم کنیم.

اجتماع دو تصویر را از حاصل جمع تصویر اصلی و مکملش بدست می آوریم. به این دلیل که ابتدا اختلاف مقدار هر پیکسل از ۲۵۵ بدست می آوریم و دوباره با مقدار پیکسلش جمع میکنیم در حقیقت اجتماع دو تصویر ماتریسی با مقدار ۲۵۵ می شود که تصویری کاملا سفید است.

خروجی:

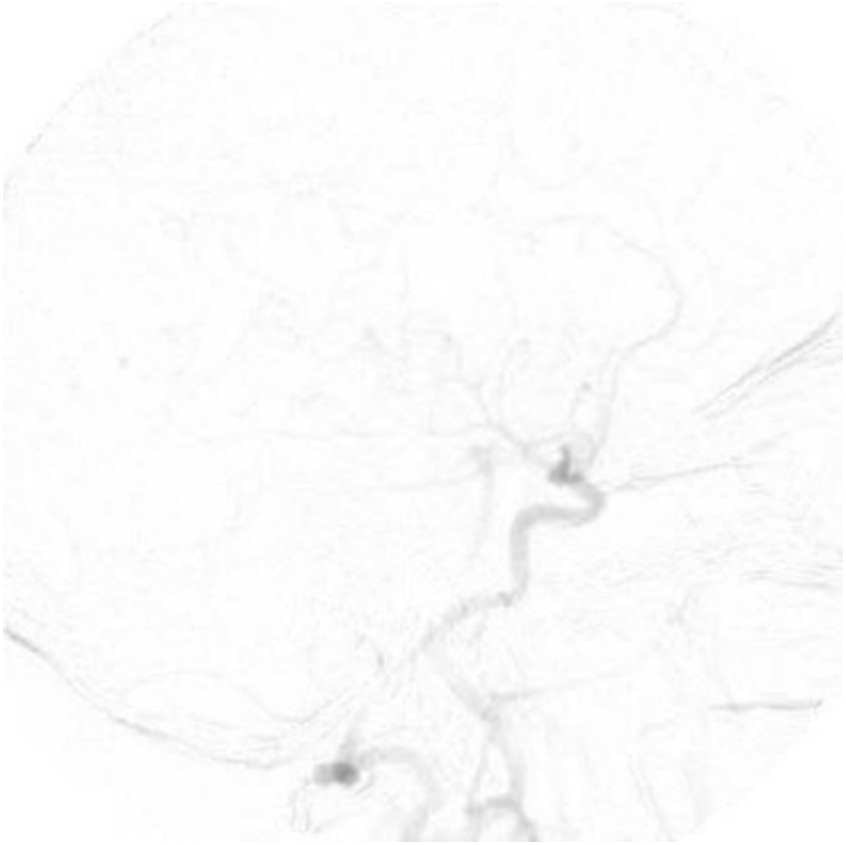


بخش ج)

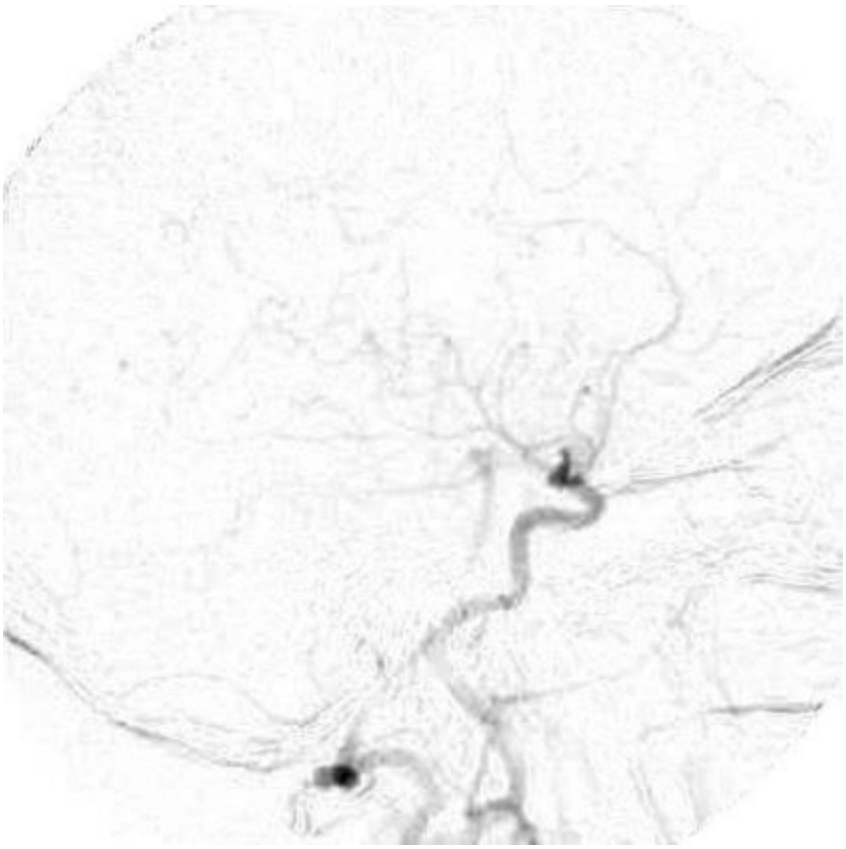
ورودی برنامه دو تصویر angiography_live.tif و angiography_mask.tif هستند. اختلاف مطلق عناصر دو تصویر را بدست می آوریم و از آن مکمل میگیریم. نرمال سازی مقادیر پیکسل ها به روش minmax حداقل مقدار در تصویر به ۰ و حداکثر مقدار نیز به ۲۵۵ می رسد . هر مقدار دیگری نیز به عددی بین ۰ و ۲۵۵ تبدیل می شود.

خروجی:

مکمل اختلاف دو تصویر



مکمل اختلاف دو تصویر(نرمال شده)



سوال شماره ۳:

ورودی برنامه تصویر استT.jpg است که با استفاده از توابع آماده میخوانیم.

Forward / backward Rotation:

در چرخاندن تصویر به روش forward ،برای هر پیکسل از تصویر ورودی مختصات جدید محاسبه میکنیم اما در چرخش تصویر به روش backward باید به شکل معکوس عمل کنیم و از تصویر forward به تصویر اصلی برسیم. برای انجام این کار از فرمول های زیر برای تعیین محل پیکسل های جدید استفاده میکنیم:

Forward:

$$i(new) = [\big((i-x)*\cos(\boldsymbol{\theta})\big) - \big((\boldsymbol{j}-\boldsymbol{y})*\boldsymbol{sin}(\boldsymbol{\theta})\big) + \boldsymbol{x}]$$

$$j(new) = [\big((i-x)*\sin(\boldsymbol{\theta})\big) + \big((\boldsymbol{j}-\boldsymbol{y})*\boldsymbol{cos}(\boldsymbol{\theta})\big) + \boldsymbol{x}]$$

Backward:

$$i(new) = [\big((i-x)*\cos(\boldsymbol{\theta})\big) + \big((\boldsymbol{j}-\boldsymbol{y})*\boldsymbol{sin}(\boldsymbol{\theta})\big) + \boldsymbol{x}]$$

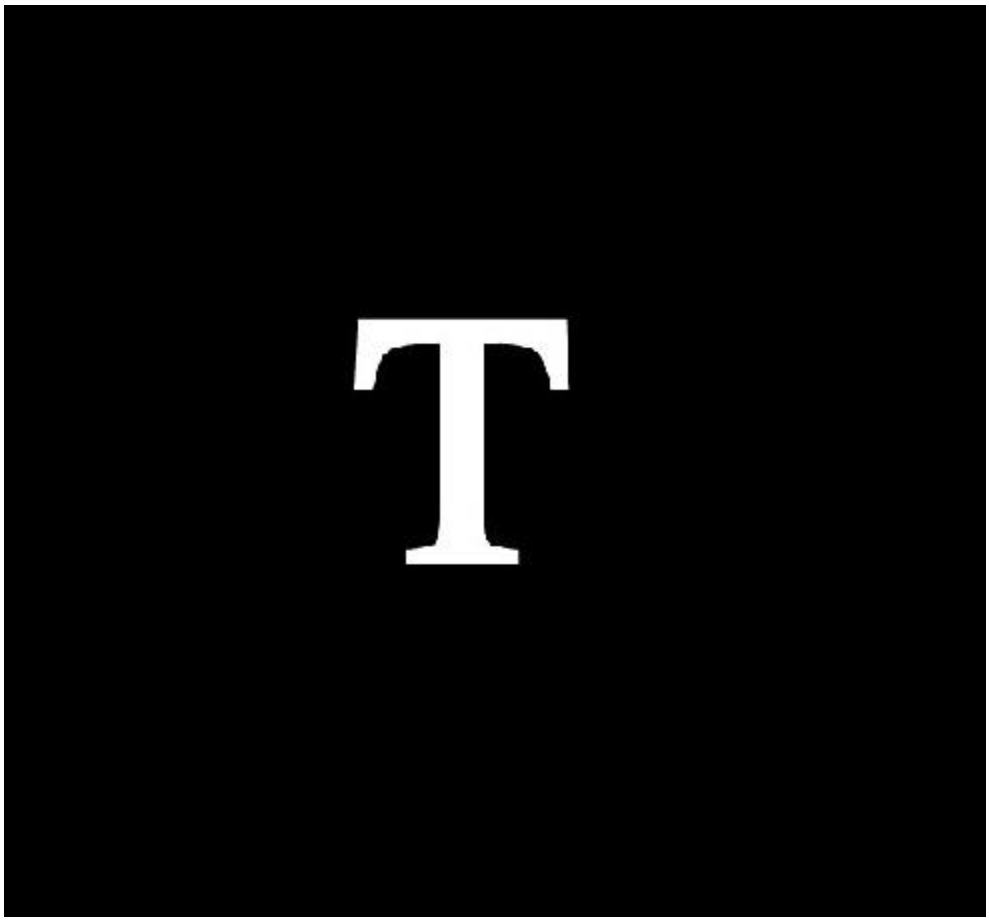
$$j(new) = [(-(i-x)*\sin(\boldsymbol{\theta})) + \big((\boldsymbol{j}-\boldsymbol{y})*\boldsymbol{cos}(\boldsymbol{\theta})\big) + \boldsymbol{x}]$$

به دلیل اینکه در این روش ممکن است محل بدست آمده صحیح نباشد آنرا در براکت قرار می دهیم . همچنین برای اینکه هنگام چرخش از محدوده خارج نشود نزدیکترین مقادیر به$\hat{x}$ و $\hat{y}$ واستفاده از آنها به عنوان مختصات مشکل را برطرف می کنیم.

خروجی:

Scaling

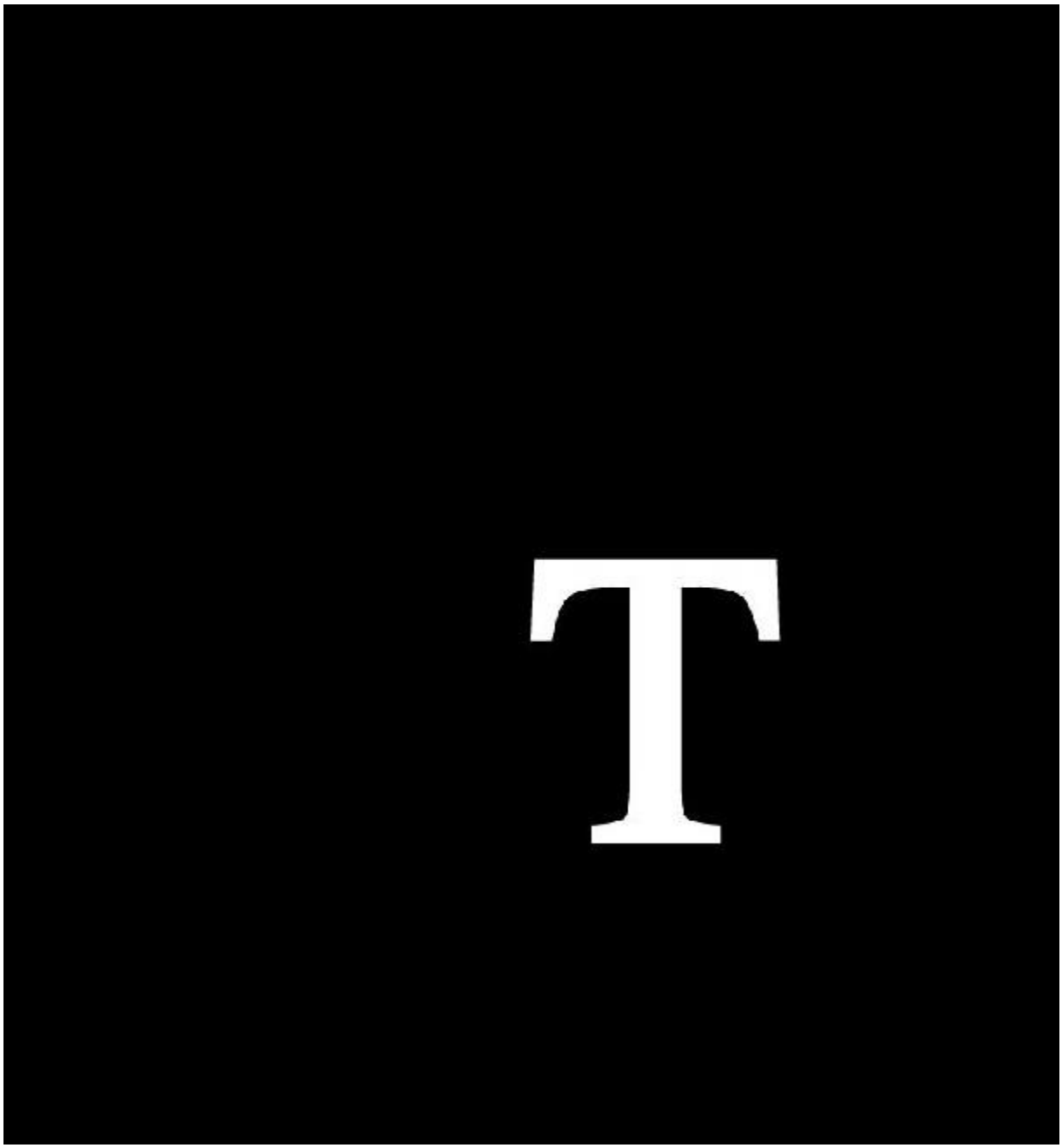
نسبت ۳/۴ برابر



Translation

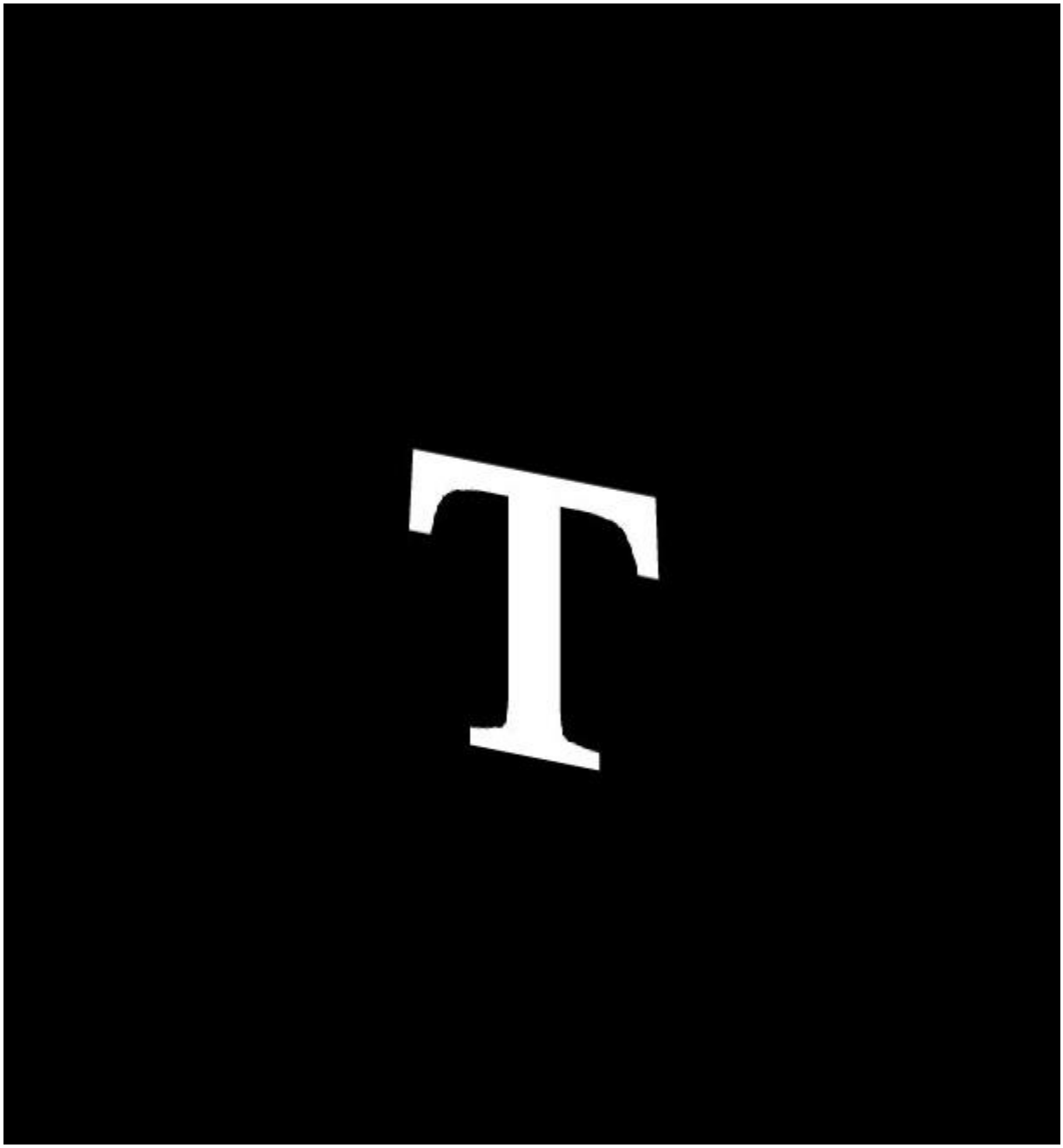
T_x=٧٠

T_y=١١٠



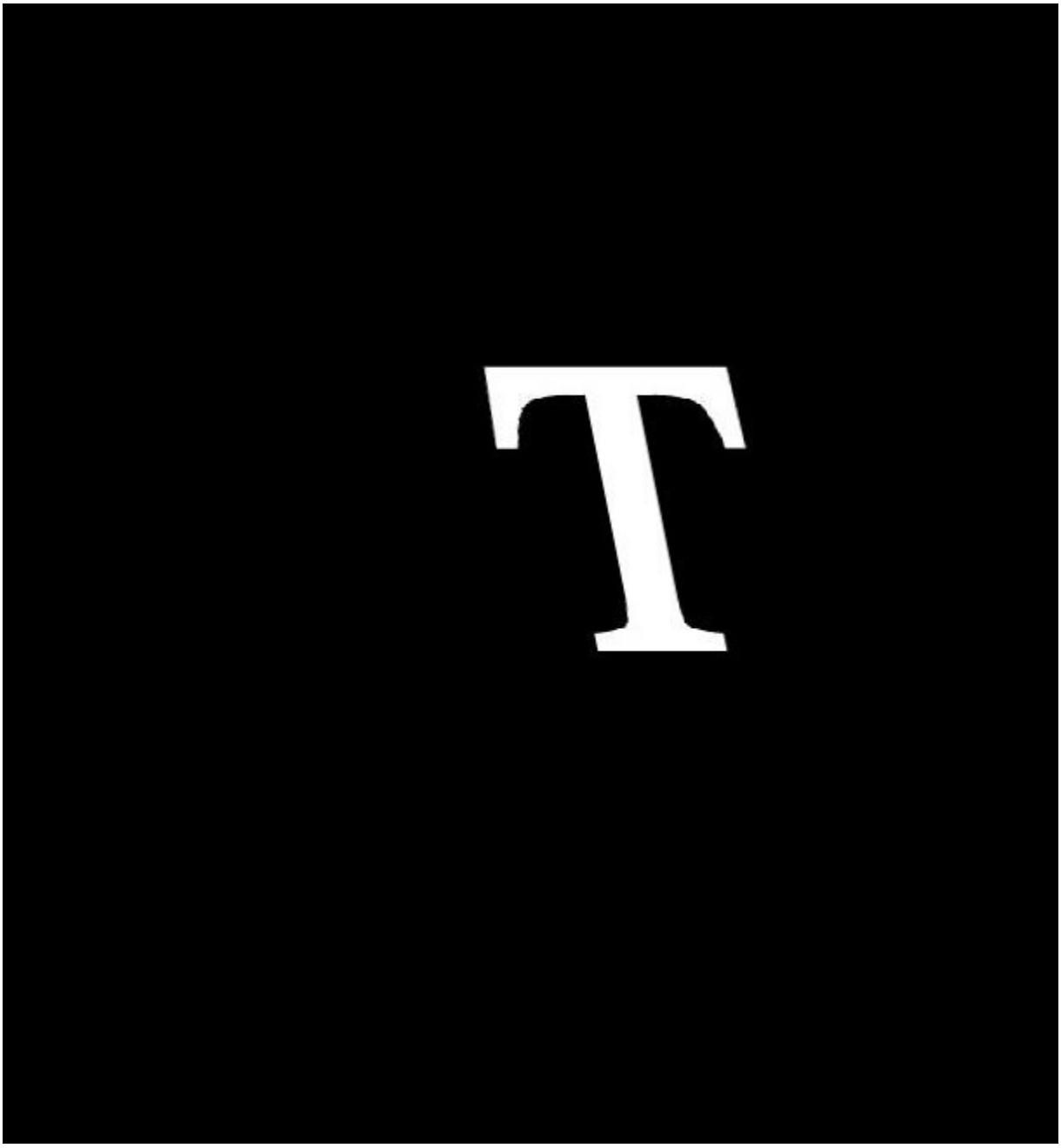
Horizontal Shear

S_h = ٠,٢



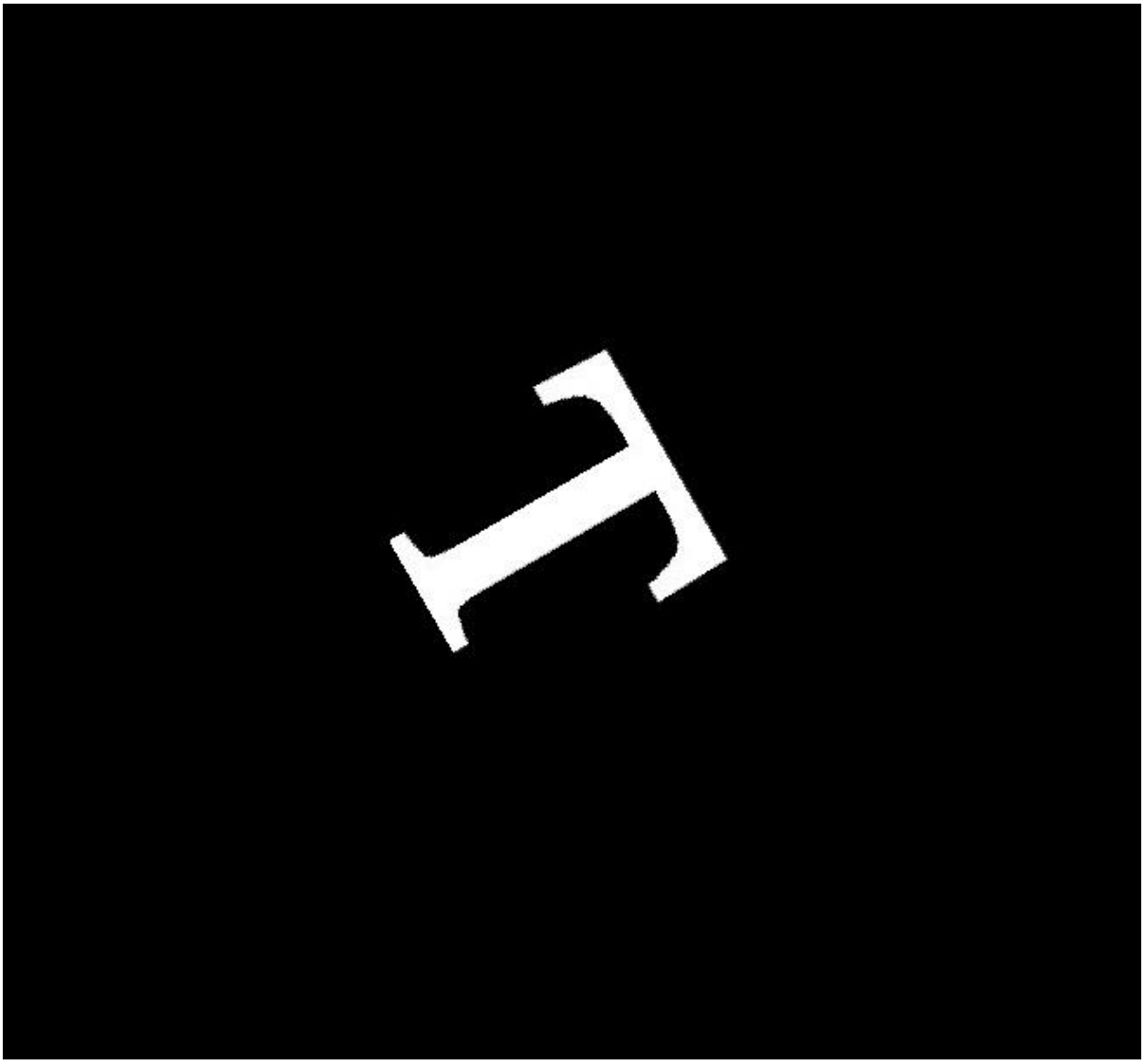
Vertical Shear

S_v = ٠,٢



Forward Rotation

$\theta = ۶۰^{\circ}$



Backward Rotation

$\theta = ۶۰°$