

به نام خدا

آزمون پایانترم درس پردازش تصویر

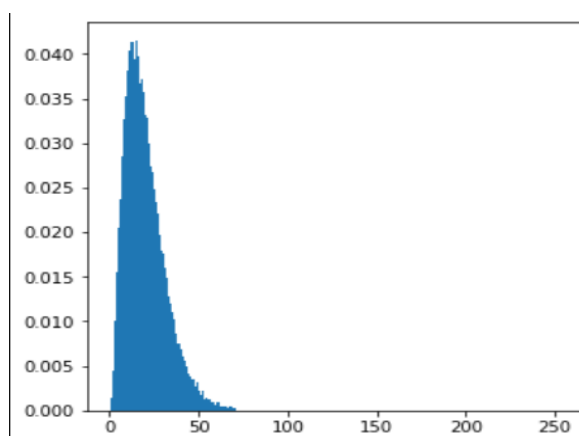
امیرحسین شریفی صدر 9733044

نام استاد : دکتر آذرنوش

سوال 1 :

الف: از آنجا که سوال از ما فراوانی نسبی، واریانس و میانگین " نویز " های موجود در تصویر را میخواهد، ما باید بخشی از تصویر را انتخاب کنیم که الگوی ثابتی را دنبال کند و ناگهان دچار تغییر **intensity** نشود از این رو ما نوار یا استریپی از تصویر را انتخاب میکنیم که این شرط برآورده شود و سپس خواسته آماری سوال را در این بخش جدا شده انجام میدهیم .

فراوانی نسبی نویز ها یا همان هیستوگرام نرمال شده



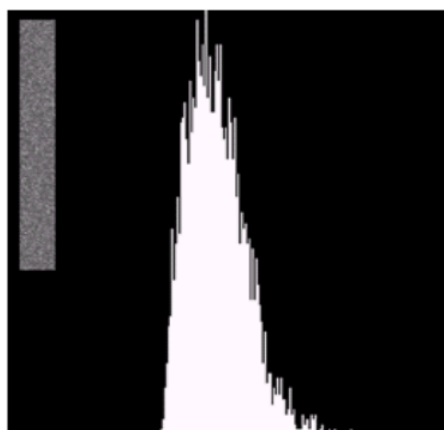
به شکل روبه ور است :

میانگین و واریانس نیز برابر

با مقادیر زیر هستند :

mean 19.88107421875
variance 127.35513400230407

ب : با توجه به نمودار فراوانی نسبی حاصل شده از نویز تصویر و مقایسه آن با جنس تابع های چگالی احتمال گفته شده در درس میتوان حدس زد جنس این تابع چگالی احتمال از نوع Rayleigh است که شکل آنرا مشاهده میکنید:



ج : فیلتر تطبیقی به این صورت است که یه پنجره را روی تصویر حرکت میدهد و واریانس نویز آن پنجره را با واریانس نویز تصویر کلی که به دست آورده بودیم مقایسه میکند . اگر واریانس نویز تصویر صفر بود یعنی نویز نداریم پس تصویر تغییری نمکند . اگر واریانس آن پنجره بسیار بزرگتر از واریانس نویز تصویر اصلی ما باشد یعنی لبه داریم و یا تغییر **intensity** بالایی ناگهان رخ داده است پس باز هم تغییری ایجاد نمیشود و همانطور که در ادامه نیز میبینیم به این علت در مرز باریکی که جمجمه را از تصویر پس زمینه جدا میکند نویز ها به شکل کامل رفع نشده اند . و در آخر اگر واریانس نویز تصویر با واریانس نویز جمجمه برابر باشد یعنی در آن بخش نویز داریم پس باید تغییری و انطباقی صورت بگیرد که در این روش میانگین آن پنجره را جایگذاری میکنیم .

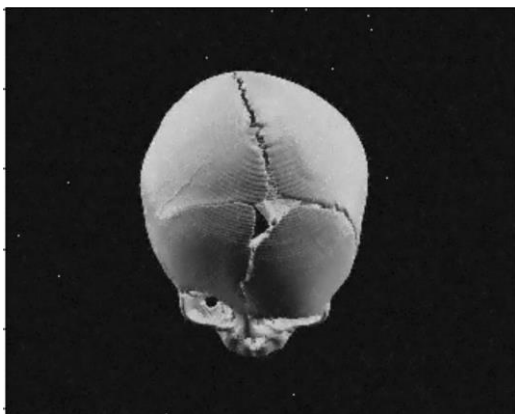
$$1) \sigma_{\eta}^2 = 0 \longrightarrow g(x, y)$$

$$2) \sigma_L^2 \gg \sigma_{\eta}^2 \longrightarrow g(x, y)$$

$$3) \sigma_L^2 = \sigma_{\eta}^2 \longrightarrow m_L$$

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_{\eta}^2}{\sigma_L^2} [g(x, y) - m_L]$$

و تصویری که با فیلتر انطباقی فیلتر شده است را مشاهده میکنید :



و تصویر **padding** شده برای این عملیات

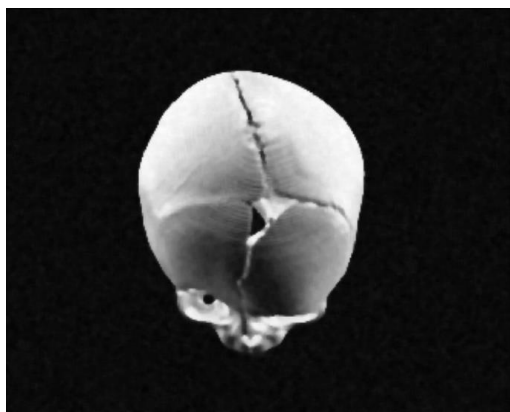
4 پیکسل در همه جهات به آن اضافه شده

است . برای اینکه بتواند پنجره های 9×9 را

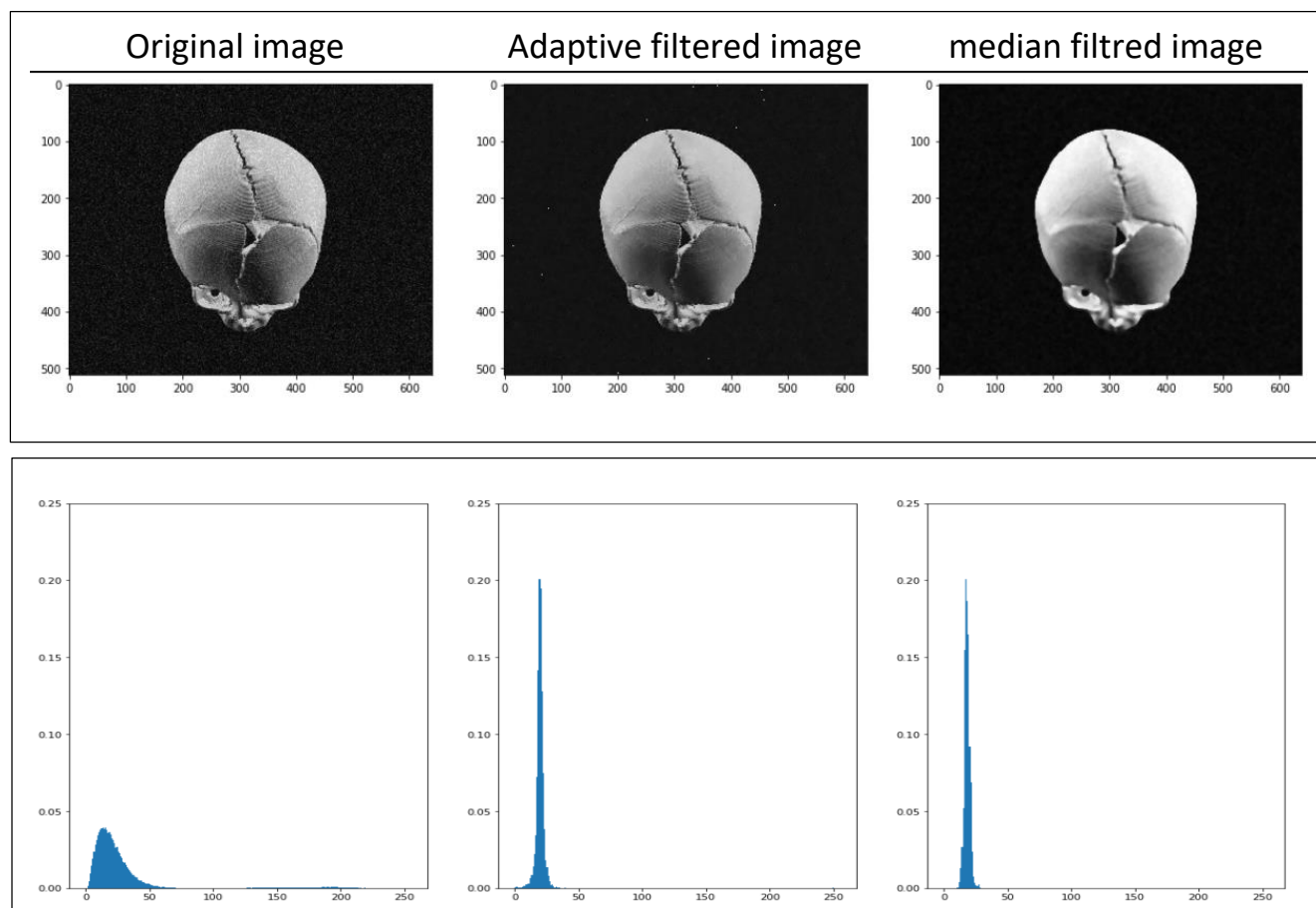
در نظر بگیرد .

د : از ما خواسته شده تا به روش میانه گیری تصویر اولیه را فیلتر کنیم که آن را با تابع آماده `cv.medianBlur` انجام می‌دهیم . نتیجه را مشاهده میکنید :

با کرنل 7×7 :



ه : نتایج را به صورت کلی همراه با هیستوگرام های نرمال شده آن ها مشاهده میکنید :



و : همانطوری که مشاهده میشود نویز تصویر تا حد خوبی از بین رفته است و در نمودار فراوانی نسبی نویز نیز نویز از گستردگی که در تصویر اصلی داشت به خطی کمی باریک تر میل کرده است .

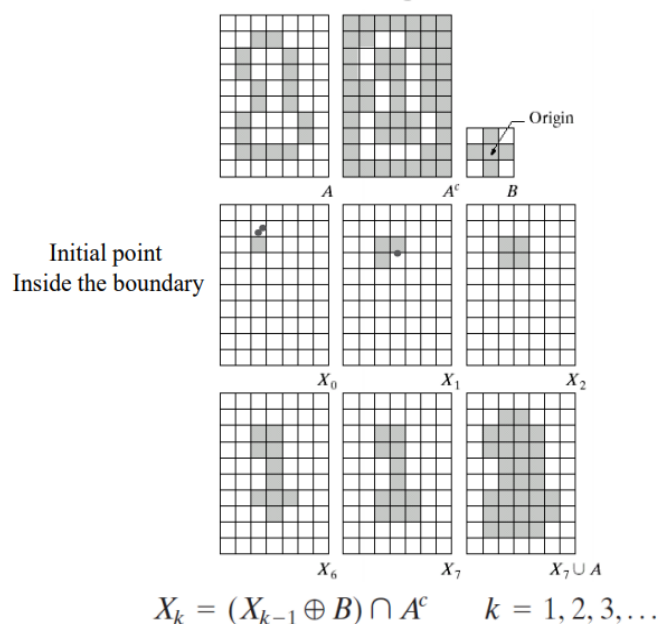
همانطور که در بخش ج الگوریتم این فیلتر را شرح دادم، بخش های از تصویر که دارای الگوی پخش یکسانی هستند و دچار نویز هستند را با جایگذاری میانگین پنجره روی آن بخش، آن محدوده را دی نویز میکنند .

ز : خیر- نتایج فیلتر میانه و فیلتر تطبیقی با هم متفاوت است زیرا فیلتر میانه جزئیات و لبه های تصویر را از بین میبرد و تصویر درست است که نویزش تا حد خوبی از بین میرود ولی بلور و بدون لبه و جزئیات میشود .

اما در فیلتر تطبیقی جزئیات تصویر حفظ میشود و لبه ها از بین نمیروند چون همانطور که نام فیلتر مشخص است سعی دارد که تصویر را تطبیق بدهد.

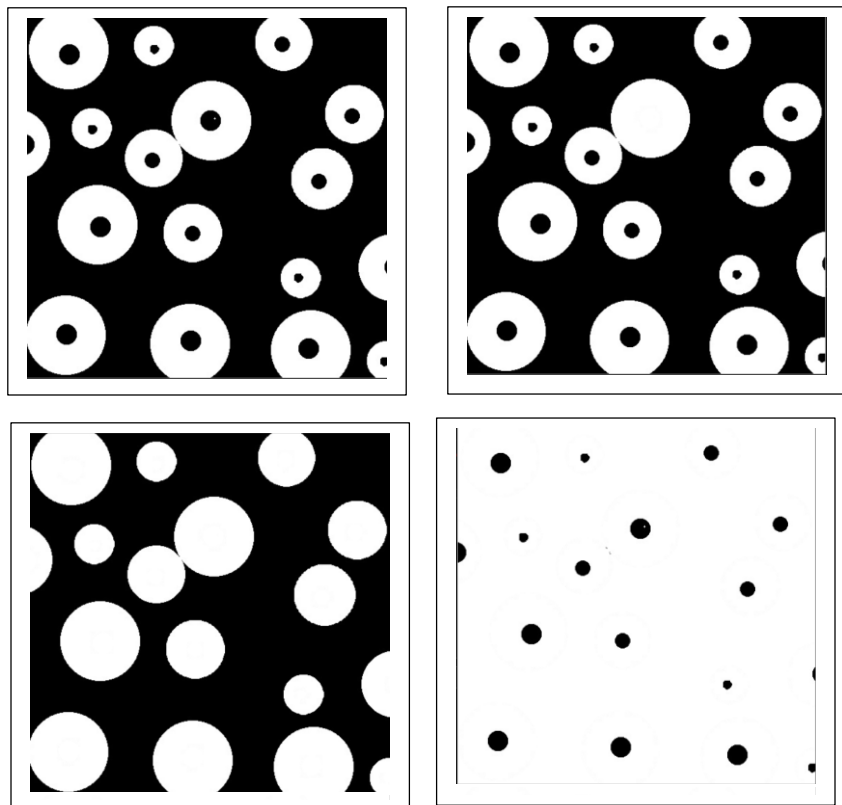
سوال 2 :

الگوریتم حل این سوال بدین شکل است که ابتدا به صورت اتوماتیک و یا توسط کاربر که در این سوال از ما با کلیک توسط کاربر خواسته شده؛ یک پیکسل را در بخشی که حفره است انتخاب میکنیم . در اینجا حفره ما سیاه است پس رنگ پیکسل را به رنگ سفید در میاوریم . سپس این پیکسل را با کرنلی به شکل "علامت جمع + " دایلیت و گسترده میکنیم اما باید حواسمان باشد که با این عمل وقتی پیکسل گسترده میشود از مرز حفره خارج نشود پس آن را با مکمل تصویر اصلی خود اشتراک میگیریم تا بخش هایی که گسترده شده اند و فراتر از حفره رفته اند حذف شوند و این عمل را تا جایی ادامه میدهیم که دیگر هر چه قدر هم عمل دایلیشن و اشتراک گیری انجام شود تغییری ایجاد نشود و حفره به طور کامل پر شود . سپس این حفره پر شده را که ما در تصویر دیگری که در ابتدا **Intensity** آن به طور کامل صفر بود انجام دادیم با تصویر اصلی خود اجتماع میگیریم که آن بخش پر شده روی حفره قرار بگیرد و حفره را کاملاً پر کند .



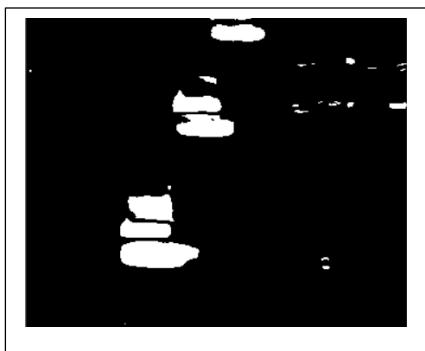
الگوریتم گفته شده در کلاس :

بنده اعمال اشتراک گیری و اجتماع گیری در سوال را ابتدا توسط فور و پیکسل به پیکسل انجام دادم که در کد به صورت کامنت مشاهده خواهید کرد و سپس برای اجرای سریع تر و بهینه تر برنامه این اعمال را توسط تابع آماده انجام دادم .



نتیجه : با کلیک بر روی هر حفره آن ها پر میشوند و اگر روی پس زمینه سیاه رنگ کلیک شود، آن به طور کل سفید میشود .

سوال 3 : روش حل این سوال بدین شکل است که میانگین تمام فریم های مو[ود در فیلم گرفته میشود تا به صورت تقریبی پس زمینه ای که ثابت است را بتوان تشخیص داد چون میانگین تمام فریم ها به پس زمینه نزدیک است . سپس اختلاف هر فریم با میانگین کل فریم ها را محاسبه کرده تا اجسام در حال حرکت درون تصویر را بتوان تشخیص داد . سپس این اختلاف را ترشولد کرده به طوری اگر اختلاف بیشتر از مقدار آستانه ای باشد آن را جسم در حال حرکت در نظر بگیرد و **Intensity** آن بخش را به مقدار 255 تغییر دهد تا به صورت واضحی تمییز شود از بخش ثابت . و اختلاف اگر کمتر از حد آستانه مشخص شده باشد آن را صفر کند که یعنی آن بخش ثابت و پس زمینه است



مقدار آستانه 50 در نظر گرفته شده

سپس **intensity** بخش های از تصویر که در تصویری که نشان دهنده اختلاف بود 255 کرده بودیم را در تصویر اصلی به رنگ قرمز، همانطور که سوال از ما خواسته در میاوریم .



سوال 4 : الگوریتم حل در این سوال که روش region growing از ما خواسته شده؛ بدین شکل است که ابتدا یک پیکسل در بخشی از تصویر انتخاب میشود (seed) سپس این پیکسل توسط معیاری با پیکسل هایی که در همسایگی اش هستند مقایسه میشود و اگر هر پیکسلی در همسایگی آن به خود پیکسل توسط معیار مشخص شده شباهت داشت آن را در بخشی ذخیره میکنیم و خود آن پیکسل یافته شده نیز مدام با همسایگی هایش چک میشود که آیا به آن ها شبیه هست یا نه که البته میتواند تمام همسایگی های هر پیکسل را با پیکسل seed یعنی اولیه خود مقایسه کرد . در نهایت میتوان توسط این الگوریتم بخشی از تصویر که به طور کلی دارای شباهت هستند را جدا کرد . و اگر در بخش های مختلف تصویر این عمل seed گذاری انجام شود میتواند تصویر را بخش بندی کرد .

در این سوال معیار ما برای مقایسه هر دو پیکسل در همسایگی هم چک کردن اختلاف Intensity آن ها با یک آستانه ترشولد مورد نظر است که در اینجا گفته شده 11 گرفته شود . یعنی اگر اختلاف Intensity دو پیکسل در همسایگی هم از 11 کمتر بود آن دو پیکسل شبیه به هم هستند و شرط similarity را برآورده میکنند .

همسایگی ها 4 خانه ای و 8 خانه ای در نظر گرفته شده اند .

برنامه نوشته شده توسط بنده به این شکل است یک seed با کلیک انتخاب میشود و intensity آن به عنوان intensity معیار مشخص میشود . آن

نقطه در در یک صفحه سیاه با اندازه تصویر اصلی با شدت 255 مشخص میکنیم. حال از اینجا شروع به چک کردن همسایگی ها و مقایسه هر کدام از آن ها با **intensity** دانه مشخص شده در تصویر اصلی میکنیم. و اگر شرط شباهت را داشت به آن تصویری که ساختیم اضافه میشود. این عمل بدین شکل است که با 2 حلقه فور از پیکسل صفر و صفر شروع میکنیم تا به اولین پیکسل با شدت 255 در تصویری که خودمان ساختیم برسیم و این پیکسل همان **seed** است. در از آنجا آغاز میکنیم به چک کردن همسایگی ها. بعد از اتمام این دو حلقه فور یک نوار از تصویر را چک کرده ایم نواری به عرض 3 پیکسل. حال در مرحله بعد باید دوباره در همین تصویر از اولیم خانه با شدن 255 شروع کنیم که این بار عوض شده. و برای اینکار باید آن دو حلقه فور را در یک وایل بگذاریم که این وایل چک میکند این عملیات تا جایی ادامه یابد که دیگر شباهتی در آن محدوده یافت نشود.

چه برای 4 همسایگی چه 8 همسایگی الگوریتم و کد یکسان هست فقط خانه هایی که باید در حلقه های ما چک شوند در 8 همسایگی بیشتر است.

و در کد یک متغیری را برابر با **True** قرار دادم که در این حالت تابع برای 4 همسایگی اجرا میشود و اگر آن **False** شود برای 8 همسایگی انجام میشود.

بخشی از بخش بندی را دیر زیر میبینید :



این تصویر با دو کلیک یعنی دو seed یکی در ناحیه
قرمز و دیگری در ناحیه آبی به دست آمده



این تصویر که تقریباً کامل بخش بندی شده است با 4 کلیک که 2
تا زانها همان نواحی شکل مقابل هستند و دتای دیگر یکی در
ناحیه بینی و دیگری در ناحیه فک هستند تشکیل شده است