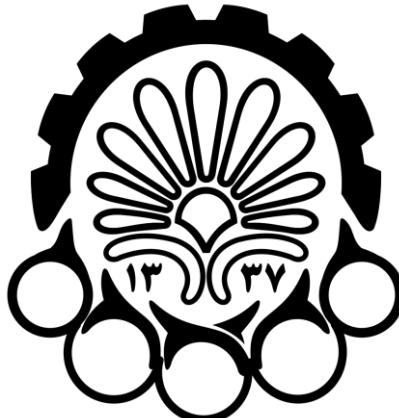


«In The Name Of GOD»



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

[HW-02-Report]

[DIGITAL IMAGE PROCESSING]

Hasan Masroor | [403131030] | April 12, 2025

## "فهرست مطالب تمرین 02"

|                 |    |
|-----------------|----|
| Problem 1 ..... | 3  |
| A) .....        | 3  |
| B) .....        | 4  |
| C) .....        | 5  |
| D) .....        | 5  |
| E) .....        | 6  |
| Problem 2 ..... | 8  |
| A) .....        | 8  |
| B) .....        | 8  |
| C) .....        | 10 |
| Problem 3 ..... | 14 |
| A) .....        | 14 |
| B) .....        | 14 |
| C) .....        | 16 |
| D) .....        | 17 |
| E) .....        | 18 |

|                 |    |
|-----------------|----|
| Problem 4 ..... | 21 |
| Problem 5 ..... | 22 |
| A) .....        | 22 |
| B) .....        | 24 |
| C) .....        | 27 |
| D) .....        | 30 |
| E) .....        | 31 |
| F) .....        | 31 |
| G) .....        | 31 |
| Problem 6 ..... | 33 |
| A) .....        | 33 |
| B) .....        | 34 |
| C) .....        | 36 |
| D) .....        | 37 |

## Problem 1: Steganography Using Bit Plane Manipulation

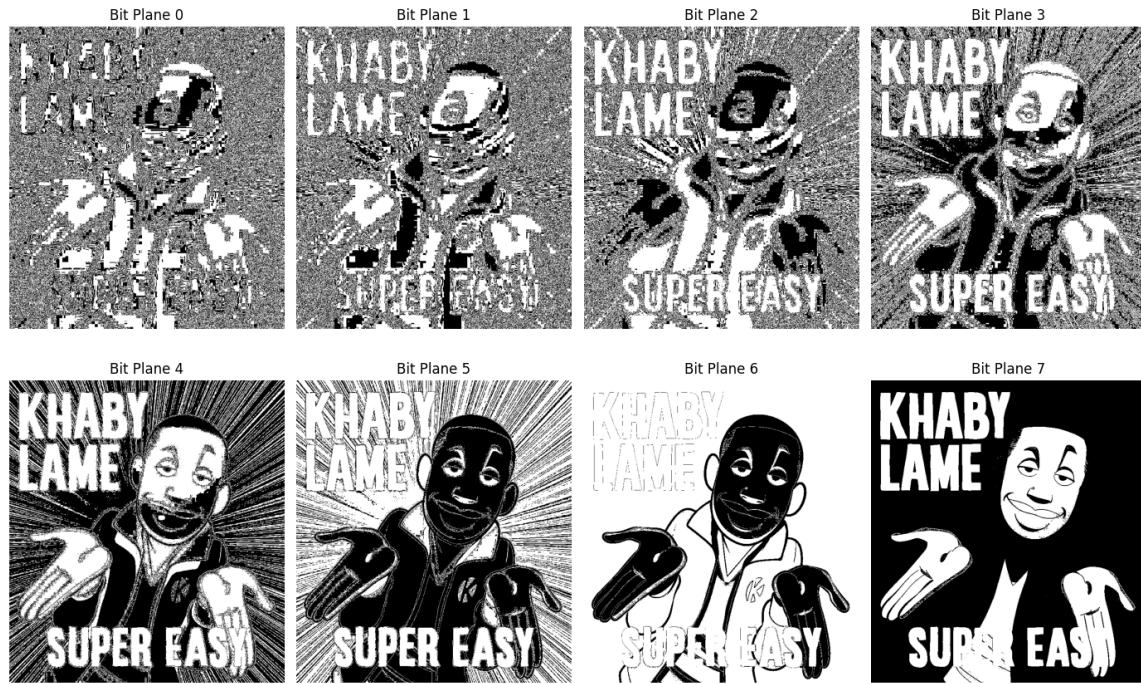
.A

در این سوال می‌خواهیم به استگانوگرافی مالتی مدیا پردازیم که به روش‌های پنهان‌سازی اطلاعات در رسانه‌های دیجیتال گفته می‌شود، به‌طوری که تغییرات ایجادشده نامحسوس باقی بمانند. در این تمرین روی روش دستکاری کم‌ارزش‌ترین بیت و تحلیل بیت پلین‌ها کار می‌کنیم.

در این پارت باید بیت پلین‌ها را استخراج کنیم و ابتدا تصویر را می‌خوانیم و به Grayscale تبدیل می‌کنیم و تصویر اصلی و این تصویر را در خروجی نمایش می‌دهیم:



حالا در کد بیت پلین‌های تصویر را استخراج می‌کنیم و ابتدا برای هر یک از 8 بیت، بیت پلین مربوطه را با استفاده از عملیات شیفت و AND محاسبه می‌کنیم؛ سپس هر بیت پلین را به صورت جداگانه نمایش می‌دهیم تا تأثیر هر بیت در تشکیل تصویر نهایی را بهتر ببینیم و خروجی‌های بیت پلین‌ها به صورت زیر هستند:



اگر به تصاویر خروجی بالا دقت کنیم می‌بینیم در بیت‌های اول ویژگی‌های دقیق‌تر تصویر هنوز خیلی واضح نیستند و به صورت نویز دیده می‌شوند (این بیت‌ها همان بین‌های کم ارزش یا LSB هستند معمولاً جزئیات خیلی زیادی ندارند). از بیت 4 به بعد به تدریج اطلاعات مهم‌تری از تصویر را می‌بینیم و بخش‌هایی از تصویر که جزئیات بیشتری دارند شروع به دیده شدن می‌کنند و در بیت 7 تصویر بهوضوح شکل اصلی خود را پیدا می‌کند و جزئیات کلیدی مثل چهره و ... را بهوضوح می‌توانیم ببینیم.

## .B

در این بخش ابتدا نام کامل‌مان را به رشتہ باینری تبدیل کرده و برای مطابقت با اندازه تصویر، رشتہ را تکرار می‌کنیم و سپس با یک حلقه for هر یک از 8 بیت پلین تصویر را با الگوی باینری حاصل از نام با استفاده از XOR ترکیب کردیم. در نهایت در خروجی هر 8 بیت پلین را که پیام خود را روی آنها اعمال کردیم نمایش می‌دهیم و اگر خوب به تصاویر پایین نگاه کنیم می‌بینیم که روی بیت پلین‌های اولیه تغییر محسوسی روی خروجی نمی‌بینیم و پیام را به خوبی قرار دادیم اما به مرور و از بیت پلین 4 به بعد کم کم تغییرات محسوسی را در خروجی می‌بینیم و در با ارزش‌ترین بیت پلین یعنی شماره 7 هم تصویر تغییرات زیادی داشته و کلا با تصویر اولیه ورودی متفاوت شده و پس نتیجه می‌گیریم که بیت پلین‌های کم ارزش برای این کار تصمیم بهتری هستند. خروجی‌های این بخش از کد به صورت زیر می‌باشد:



## .C

در این پارت نیز باید پیام پنهان را از روی یکی از بیت‌پلین‌های مرحله قبل استخراج کنیم و به همین منظور بیت پلین 2 را به عنوان نمونه انتخاب کردیم و با استفاده از XOR بین بیت‌پلین تغییریافته و بیت‌های باینری پیام، بیت‌های مخفی را استخراج می‌کنیم و سپس این بیت‌ها را به رشته باینری تبدیل کردیم و در نهایت در خروجی هم رشته باینری پیام و هم خود پیام را نمایش دادیم:

Binary String: 01101000011000010111001101100001011011100010000001101101 ...

Hidden Message: hasan masroor

## .D

در این پارت برخلاف قسمت قبلی باید بدون اطلاع داشتن از طول پیام، پیام مخفی را استخراج کنیم. به همین منظور مثل قبل ما پیام را به باینری تبدیل کردیم اما برخلاف پارت قبل می‌آییم و یک مارکر پایانی نیز به آن اضافه می‌کنیم تا مشخص شود که پیام کجا تمام می‌شود. در ادامه با تابعی که در کد تعریف کردیم با استفاده از XOR بین بیت‌های تغییریافته و اصلی داده‌های مخفی را استخراج کردیم و برای فهمیدن پایان پیام هم وقتی مارکر پایانی را

تشخیص دادیم متوجه می‌شویم که طول پیام تمام شده و متوقف می‌شویم و در نتیجه بدون اطلاع داشتن از طول پیام می‌توانیم پیام مخفی را استخراج کنیم و خروجی این بخش را به صورت زیر داریم:

Extracted Name: hasan masroor

.E

حالا در این پارت باید به ترید آف بین غیرقابل تشخیص بودن، مقاوم بودن و ظرفیت در استگانوگرافی بین پلین‌ها پردازیم اما قبل از آن برا تحلیل بهتر مجدد خروجی پارت b را که پیام مخفی را روی هر یک از بین پلین‌ها اعمال کردیم نمایش می‌دهیم:



#### :Imperceptibility ▪

اگر به تصاویر بالا دقت کنیم می‌بینیم که تغییرات در بیت‌های کم ارزش نامحسوس‌تر است (به خاطر اینکه این بیت‌ها شامل جزئیات و نواحی با دقت پایین و رودی هستند) و مثلاً در بیت پلین 0 تغییرات اعمال شده را با چشم نمی‌توانیم تشخیص دهیم و همچنان شبیه تصویر اولیه ما است؛ اما هرجه ارزش این بیت‌ها بالاتر رود و به سمت بیت‌های با ارزش برویم تغییرات اعمال شده محسوس تر می‌شوند و به راحتی می‌توان آنها را تشخیص داد و در نتیجه غیرقابل تشخیص بودن آنها کمتر می‌شوند.

### **:Robustness ▪**

از نظر سطح مقاوم بودن (مقاوم بودن داده‌های مخفی در برابر عوامل مختلف) بیت پلین‌هایی که در سطح پایین‌تری قرار دارند مقاومت کمتری دارند و برای مقاومت بیشتر معمولاً از بیت‌های بالاتر می‌توان استفاده کرد که در برابر عوامل مختلف مثل نویز، فشرده‌سازی و دستکاری‌های مختلف مقاومت بیشتری دارند اما همانطور که در سوال هم به آن اشاره شده یک ترید آف بین این ویژگی‌ها داریم و مثلاً همین مقاومت بیشتر بیت پلین‌ها هزینه دارد و تغییرات در بیت‌های بالاتر از آنجایی که این بیت‌ها جزئیات بیشتری دارند می‌تواند باعث آسیب به کیفیت تصویر شود.

### **:Capacity ▪**

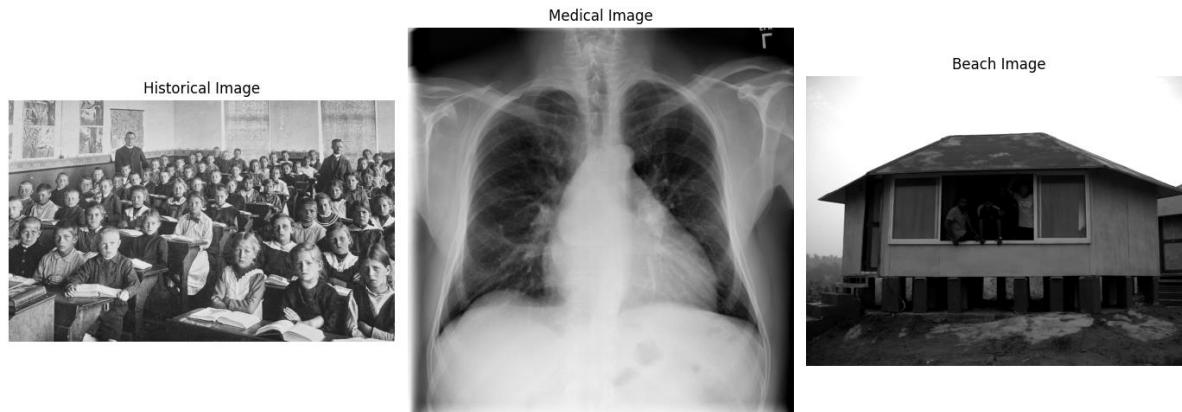
همان‌طور که قبل‌گفتیم بیت پلین‌های کم ارزش‌تر (چون جزئیات و دقت کمتری از تصویر اصلی را دارا هستند) معمولاً ظرفیت بیشتری برای جاسازی داده‌ها دارند و می‌توان داده‌های بیشتری را در آنها ذخیره کرد بدون اینکه تغییرات زیاد و محسوسی روی تصویر اصلی ما بگذارند اما در بیت پلین‌های با ارزش‌تر (چون جزئیات بیشتر و دقیق‌تری از تصویر اصلی را داریم) معمولاً ظرفیت کمتری داریم و از بخش محدودی از پیکسل‌ها می‌توانیم استفاده کنیم و داده‌های زیاد باعث از دست‌رفتن اطلاعات بیشتری از تصویر می‌شوند.

پس در مجموع وقتی که بخواهیم بین کیفیت تصویر، مقاومت پیام، و مقدار اطلاعاتی که می‌خواهیم مخفی کنیم تعادل برقرار کنیم، باید این مورد را در نظر بگیریم که هر بیت‌پلین ویژگی‌های مختص به خود را دارد و بیت‌های پایین‌تر برای بهتر مخفی کردن پیام گزینه بهتری هستند؛ زیرا تغییرات محسوسی در ظاهر تصویر ایجاد نمی‌کنند اما خیلی مقاوم نیستند. از آن طرف بیت‌های بالاتر و با ارش‌تر قوی‌تر هستند اما ممکن است تصویر را خراب کنند و باعث آسیب به کیفیت تصویر شوند؛ پس در نهایت بسته به هدفی که برای مخفی کردن پیام داریم باید درست و هوشمندانه تصمیم بگیریم که از کدام بیت استفاده کنیم و چه تعادلی در رابطه با ویژگی‌های گفته شده را در نظر بگیریم.

## Problem 2: Advanced Histogram-Based Image Enhancement Techniques

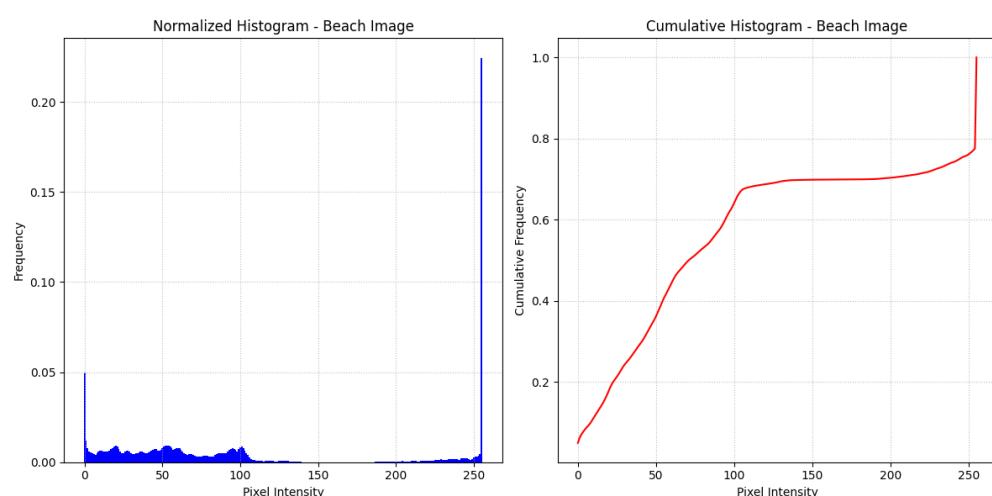
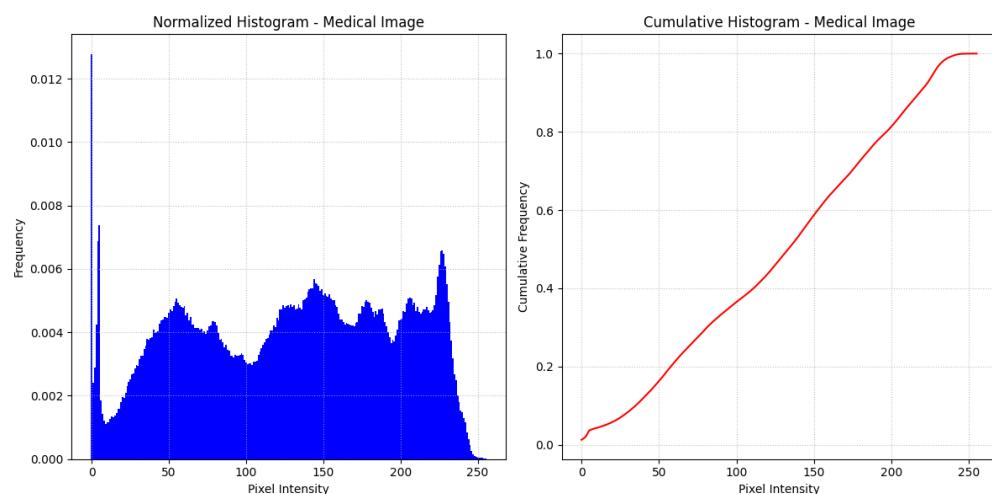
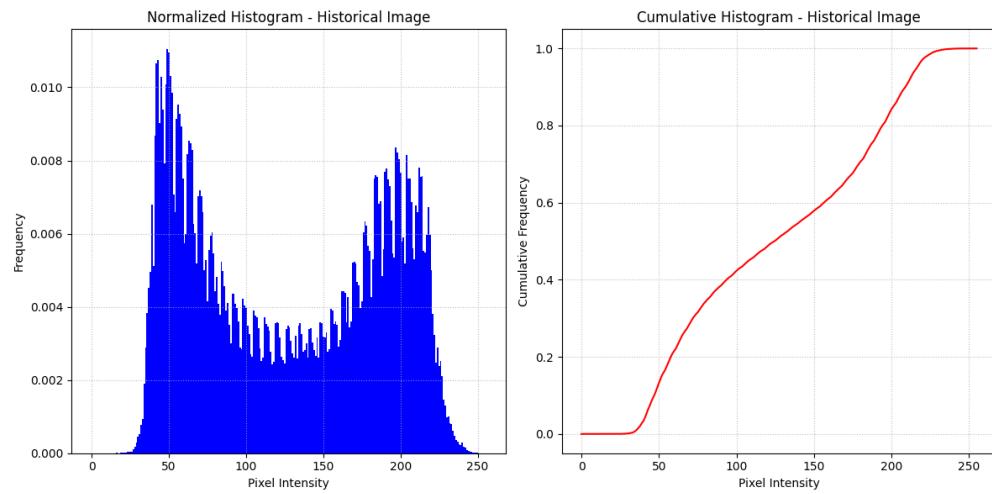
.A

در این تمرین روی یکسان‌سازی هیستوگرام و چهار روش HE، BBHE، BPDHE و DSIHE برای بهبود کنترast تصاویر داده شده قرار هست کار کنیم. ابتدا کتابخانه‌های مورد نیاز را import می‌کنیم و سپس تصاویر را به فرم تبدیل می‌کنیم و این سه تصویر را در خروجی نمایش می‌دهیم:



.B

حالا در این پارت باید هیستوگرام (نمایش تعداد پیکسل‌ها برای هر سطح روشنایی) تصاویر قبل و همچنین هیستوگرام تجمعی (مجموع مقادیر هیستوگرام که نشان می‌دهد تا هر سطح روشنایی چند پیکسل در تصویر وجود دارد) را محاسبه کنیم و نمایش دهیم. در کدی که برای این بخش زدیم ابتدا برای هر تصویر هیستوگرام نرمال‌شده را محاسبه کردیم تا بتوانیم مقایسه بهتری بین تصاویر داشته باشیم (نرمال‌سازی باعث می‌شود مقادیر هیستوگرام در بازه 0 تا 1 قرار گیرند و مستقل از اندازه تصاویر بتوانیم مقایسه بصری دقیق‌تر و بهتری داشته باشیم)؛ سپس با استفاده از مجموع تجمعی مقادیر هیستوگرام، هیستوگرام تجمعی را محاسبه کردیم و در نهایت نیز هم هیستوگرام نرمال شده و هم هیستوگرام تجمعی را برای هریک از تصاویر ورودی نمایش دادیم:



به عنوان مثال اگر هیستوگرام‌های مربوط به تصویر پزشکی نگاه کنیم، در هیستوگرام چپی که مربوط به هیستوگرام نرمال شده است می‌بینیم که پیک‌های بالا در نزدیکی شدت‌های پایین (حدود ۰) و شدت‌های بالا (حدود ۲۳۰) نشان می‌دهد که هم نواحی بسیار تیره و هم نواحی بسیار روشن در تصویر پزشکی مربوطه داریم؛ همچنین این نمودار نشان می‌دهد که تصویر دارای توزیع نسبتاً گستردگی از شدت‌های پیکسلی مختلف است و چندین پیک در نواحی مختلف دیده می‌شود که نشان‌دهنده‌ی وجود چند ساختار یا بافت مختلف در تصویر پزشکی است. نمودار راستی نیز که برای هیستوگرام تجمعی است نشان می‌دهد که درصد تجمعی پیکسل‌ها تا یک شدت مشخص چقدر است. شکل منحنی تقریباً خطی با شیب متغیر است که نشان می‌دهد تصویر توزیع نسبتاً معادلی از شدت‌ها دارد (هرجا شیب منحنی بیشتر باشد یعنی در آن بازه تعداد بیشتری پیکسل با آن شدت‌ها وجود دارد و بخش‌های صاف‌تر از منحنی نیز نشان‌دهنده بازه‌هایی از شدت هستند که پیکسل زیادی ندارند).

برای هیستوگرام‌های دو تصویر دیگر نیز می‌توانیم به همین صورت تجزیه و تحلیل کنیم و مثلاً برای تصویر اول تقریباً مثل تحلیل تصویر پزشکی است و دارای توزیع نسبتاً گستردگی است و برای بعضی شدت‌ها مثل ۰ پیکسلی نداشته و برای تصویر سوم هم می‌بینیم در نزدیکی رنگ مشکی شیب بسیار زیادی گرفته است و در این شدت پیکسل‌های زیادی داشته است.

## C.

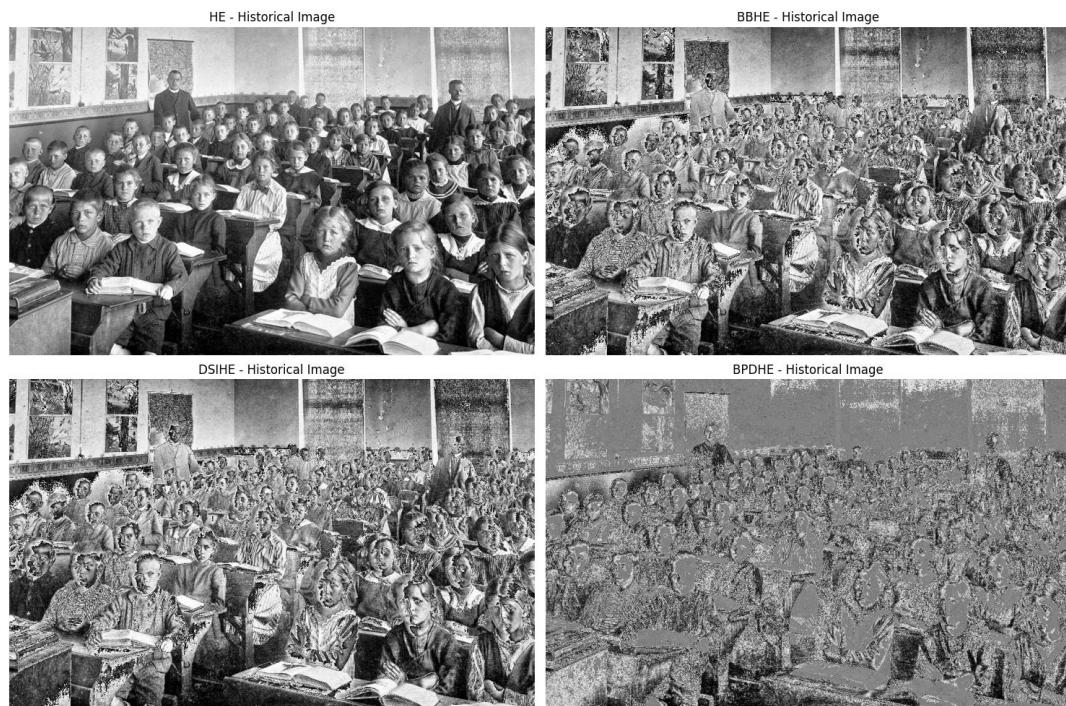
در این بخش باید برای بهبود کنتراست و روشنایی تصاویر از چهار روش یکسان‌سازی (معادل‌سازی یا یکنواخت‌سازی) هیستوگرام استفاده کنیم. ابتدا با استفاده از HE یک تبدیل ساده برای یکسان‌سازی هیستوگرام و بهبود کنتراست اعمال شد؛ سپس در BBHE با تقسیم تصاویر به دو ناحیه بر اساس روشنایی میانگین و اعمال HE به طور جداگانه، تلاش کردیم تا حفظ روشنایی در تصویر بهبود یابد. در روش DSIEH نیز به جای میانگین روشنایی از یک آستانه برای تقسیم هیستوگرام به دو بخش متوازن استفاده شد و سپس HE را به هر بخش اعمال کردیم. در نهایت در BPDHE از فیلتر گاوی برای اسموثینگ هیستوگرام استفاده دو پس از شناسایی پیک‌های محلی، هیستوگرام به بخش‌های مختلف تقسیم و به هر بخش HE اعمال شد.

برای ارزیابی عملکرد این تکنیک‌ها طبق سوال از سه معیار گفته شده یعنی PSNR، SSIM و MBS استفاده می‌کنیم. PSNR نشان‌دهنده نسبت سیگنال به نویز در تصویر است و بالاتر بودن آن نشان‌دهنده کیفیت بهتر تصویر است، SSIM میزان شباهت ساختاری بین تصویر اصلی و تصویر بهبود یافته را ارزیابی می‌کند و هرچه مقدارش بیشتر باشد به این معناست که تصویر بهبود یافته از نظر ساختاری بیشتر شبیه تصویر اصلی است و MBS نیز تغییرات متوسط روشنایی تصویر را پس از بهبود کنتراست اندازه‌گیری می‌کند و هرچه مقدارش پایین‌تر باشد نشان‌دهنده حفظ بهتر روشنایی تصویر اصلی است.

در ادامه هم خروجی اعمال ۴ تکنیک گفته شده و همچنین مقادیر این معیارها را تا ۴ رقم نشان دادیم و به تجزیه و تحلیل آنها می‌پردازیم.

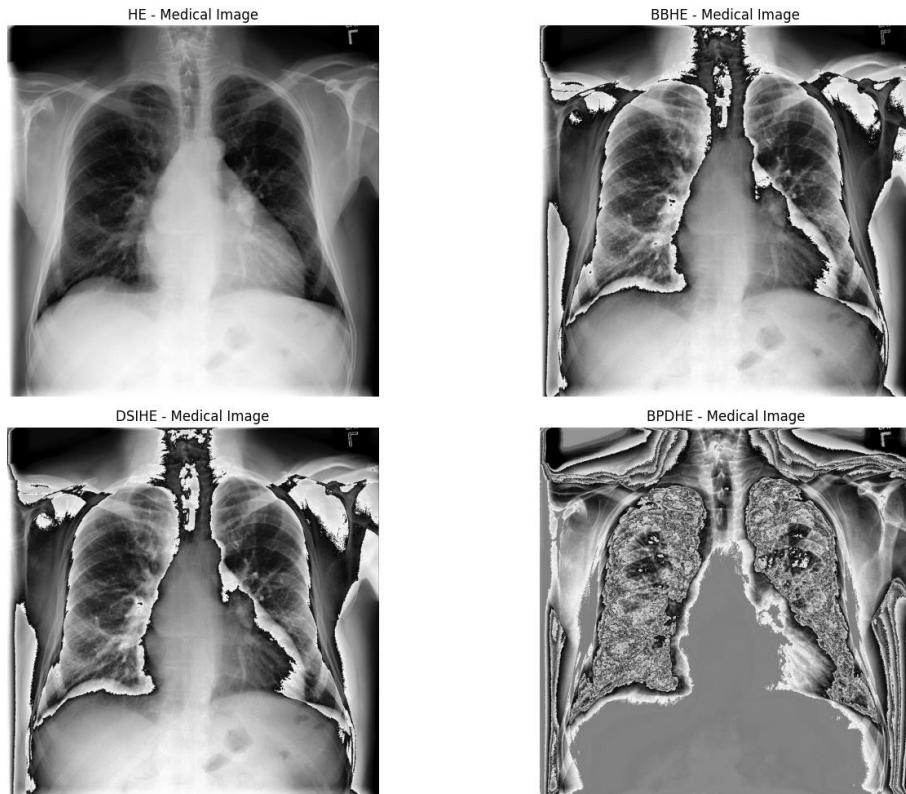
خروجی سه معیار گفته شده برای تصویر historical image به صورت زیر است و می بینیم که تکنیک HE با PSNR برابر 25.1472 نشان دهنده کیفیت خوب تصویر بهبود یافته است که جزئیات تصویر حفظ شده اند، SSIM با مقدار 0.9050 نیز نشان می دهد که ساختار اصلی تصویر به خوبی حفظ شده است و همچنین MBS با مقدار 0.9050 نیز به این معنی است که تغییرات معمولی در روشنایی تصویر ایجاد شده و تکنیک HE توانسته به خوبی روشنایی اصلی تصویر را حفظ کند و اگر به تصویر خروجی که پایین همین جدول نشان داده شده است نگاه کنیم این موضوع را به خوبی می توانیم ببینیم. در سه تکنیک بعدی می بینیم مقدار PSNR کاهش یافته و به تبع آن کیفیت تصویر بهبود یافته نیز کاهش یافته است و کاهش کیفیت در این سه تکنیک در تصویر پایین مشخص هستند و می بینیم HE کیفیت بالاتری دارد. در تکنیک BBHE و DSIHE می بینیم که مقدار SSIM کم شده است و نشان می دهد که ساختار تصویر به شدت تغییر کرده است و در DHE نیز مقدار آن کاهش بیشتری پیدا کرده است و اگر به تصویر مربوطه نگاه کنیم می بینیم که تغییرات شدیدی در تصویر ایاد شده است و جزئیات اصلی آن از دست رفته است. همانطور که می بینیم MBS تکنیک دوم و سوم تقریباً نزدیک به هم هستند نسبت به بقیه تکنیک ها نتوانستند روشنایی تصویر را به خوبی حفظ کنند و در نهایت برای BPDHE کمترین مقدار را داشته و به همین سبب بهترین حفظ روشنایی را در بین این روش ها داشته است:

| Historical Image Evaluation |         |        |        |
|-----------------------------|---------|--------|--------|
| Method                      | PSNR    | SSIM   | MBS    |
| HE                          | 25.1472 | 0.9050 | 0.9095 |
| BBHE                        | 10.6761 | 0.3464 | 1.6410 |
| DSIHE                       | 10.6775 | 0.3469 | 1.6650 |
| BPDHE                       | 10.5619 | 0.1070 | 0.4952 |



خروجی سه معیار گفته شده برای تصویر medical image نیز به صورت زیر است و می‌توانیم نتایج بالا اینها را هم کامل و جامع تجزیه و تحلیل کنیم اما با توجه به جدول پایین و تصویر خروجی این ۴ تکنیک می‌بینیم که مثل قبلي روشن HE بيشترین مقدار برای دو معیار PSNR و SSIM دارد و به تصویر مربوطه هم نگاه کنیم مشخص است که کیفیت و جزئیات بیشتری از تصویر نسبت به سایر روش‌ها قابل مشاهده است. از آن طرف روش BPDHE نیز کمترین مقدار MBS را داشته که بهترین حفظ روشنایی بین سایر روش‌ها محسوب می‌شود:

| Medical Image Evaluation |         |        |        |
|--------------------------|---------|--------|--------|
| Method                   | PSNR    | SSIM   | MBS    |
| HE                       | 29.2342 | 0.9773 | 2.1893 |
| BBHE                     | 11.3254 | 0.6724 | 1.5999 |
| DSIHE                    | 11.3567 | 0.6778 | 1.6373 |
| BPDHE                    | 9.9393  | 0.3770 | 0.3887 |



خروجی سه معیار گفته شده برای تصویر beach image نیز به صورت زیر است و با توجه به جدول پایین و تصویر خروجی این ۴ تکنیک می‌بینیم که مثل قبلي روشن HE بيشترین مقدار برای دو معیار PSNR و SSIM دارد و به تصویر مربوطه هم نگاه کنیم مشخص است که کیفیت و جزئیات بیشتری از تصویر نسبت به سایر روش‌ها قابل مشاهده است. در رابطه با معیار MBS می‌بینیم سه تکنیک اول عملکرد خوبی از خود نشان ندادند و در BBHE تغییرات روشنایی خیلی زیادی داریم اما مجدد BPDHE کمترین مقدار را به خود اختصاص داد است و در حفظ روشنایی بهترین عملکرد را از خود نشان داده است.

| Beach Image Evaluation |         |        |         |
|------------------------|---------|--------|---------|
| Method                 | PSNR    | SSIM   | MBS     |
| HE                     | 17.5394 | 0.8641 | 17.8058 |
| BBHE                   | 9.1937  | 0.5992 | 31.6364 |
| DSIHE                  | 9.9248  | 0.4962 | 24.6710 |
| BPDHE                  | 10.2845 | 0.3849 | 0.3965  |



برای تصاویر رادیوگرافی پزشکی، روش BPDHE به دلیل حفظ بهتر روشنایی در تصاویر گزینه مناسب‌تری است. در این نوع تصاویر حفظ دقت روشنایی برای تشخیص‌های پزشکی اهمیت زیادی دارد حتی اگر کیفیت کلی تصویر کمی کاهش یابد. در حالی که روش HE می‌تواند کیفیت بالاتری از نظر مقایسه کلی تصویر ارائه دهد اما از آن طرف تغییرات شدید در روشنایی آن می‌تواند منجر به اشتباهات در تفسیر پزشکی شود؛ بنابراین برای استفاده در تصاویر رادیوگرافی، BPDHE به دلیل اولویت دادن به حفظ روشنایی بهتر عمل می‌کند و انتخاب بهتری است.

برای بازسازی تصاویر تاریخی روش HE بهترین انتخاب است. این روش توانسته به خوبی تعادل بین کیفیت تصویر و حفظ روشنایی برقرار کند که برای بازسازی تصاویر تاریخی اهمیت زیادی دارد؛ در حالی که روش‌های دیگر مانند DSIHE و BBHE عملکرد ضعیفت‌تری در حفظ کیفیت و روشنایی دارند؛ پس HE بهترین انتخاب برای این نوع تصاویر است و عملکرد بهتری دارد.

### Problem 3: Create Your Own Pointillism Art

.A

در این تمرین باید یک اثر هنری به سبک پوینتیلیسم (نقاشی با نقطه‌های رنگ) ایجاد کنیم. برای این کار باید ابتدا لبه‌های تصویر را با فیلترهای مکانی استخراج کنیم و سپس تصویر نهایی را با جایگذاری تصادفی نقطه‌های رنگی بر اساس لبه‌ها، رنگ‌های تصویر اصلی و ... بازسازی کنیم که در نهایت خروجی یک تصویر با سبک پوینتیلیسم سبک خواهد بود.

ابتدا کتابخانه‌های مورد نظر را `import` می‌کنیم، بعد از اینکه تصاویر تمرین را آپلود کردیم به فرم RGB و `Grayscale` تبدیل می‌کنیم و در نهایت در خروجی نمایش‌شان می‌دهیم:

.B

بعد از اینکه در پارت قبل تصاویر را نمایش دادیم در این بخش باید سه فیلتر تشخیص لبه که در سوال گفته را پیاده‌سازی کنیم و روی تصاویر اعمال کنیم. فیلتر Sobel و Prewitt را هم در جهت X و هم در جهت Y محاسبه می‌کنیم و آنها را با هم ترکیب می‌کنیم و فیلتر Laplacian را هم برای تشخیص لبه‌های تیز به کار ببریم (از کرنل با سایز 3 برای این فیلترها استفاده کردیم).

ابتدا فیلتر Sobel را به صورت جداگانه هم در جهت X و هم در جهت Y اعمال می‌کنیم که این فیلتر تغییرات شدت روشنایی را در هر دو جهت شناسایی می‌کند؛ سپس نتایج این دو فیلتر را ترکیب کردیم (به این صورت که مجموع مربعات رو جهت را حساب کردیم و از آن جذر گرفتیم). ماتریس Sobel برای لبه‌های افقی و عمودی به صورت زیر هستند:

$$\text{Sobel} \quad h_{S_x} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad \text{and} \quad h_{S_y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

سپس فیلتر Laplacian را برای شناسایی لبه‌های تیز در تصویر به کار بردیم. ماتریس Laplacian را هم می‌توانیم به یکی از صورت‌های زیر استفاده کنیم که سمت چپ پیش فرض در OpenCV می‌باشد و سمت راستی را هم توانیم در صورت نیاز خودمتن تنظیم کنیم:

3-by-3  
Laplacians

$$h_{L1} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \text{and} \quad h_{L2} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

مشابه فیلتر Sobel ، فیلتر Prewitt نیز برای شناسایی لبه‌ها در دو جهت X و Y اعمال هستند و برای اعمال فیلتر Prewitt ، ماتریس‌هایی برای جهت‌های X و Y که جلوتر نشان می‌دهیم تعریف کردیم و مثل Sobel در نهایت نتایج را با هم ترکیب کردیم. ماتریس Prewitt برای لبه‌های افقی و عمودی به صورت زیر هستند:

Prewitt

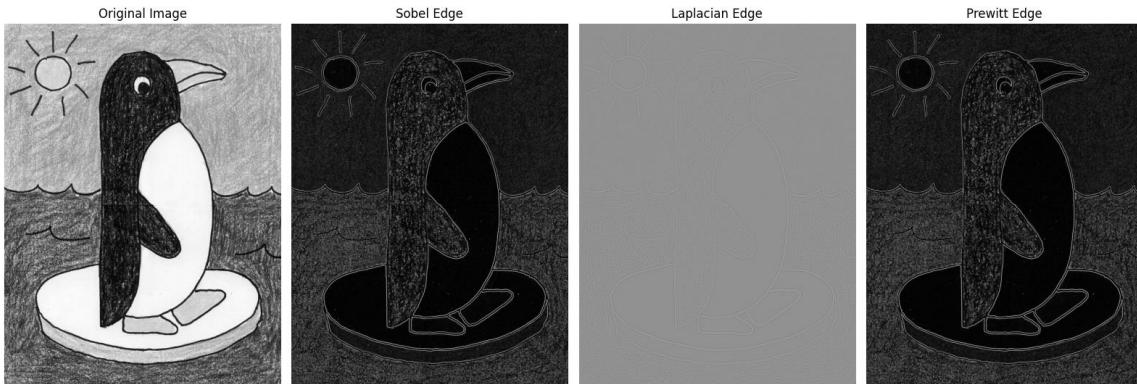
$$h_{Px} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad \text{and} \quad h_{Py} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix}$$

در نهایت نیز نتایج هر سه فیلتر به صورت یک دیکشنری بازگشت داده می‌شوند که شامل تصاویر لبه‌های شناسایی شده با هر فیلتر است و نتایج حاصل از اعمال سه فیلتر روی تصویر پرتره و نقاشی پنگوئن را در خروجی نمایش می‌دهیم:

Edge Detection for Image 1 - Jan Gossaert: Self-portrait



## Edge Detection for Image 2 - Penguin Drawing



به عنوان مثال از خروجی‌های بالا نتایج تصویر پرتره را تجزیه و تحلیل می‌کنیم و مشاهده می‌کنیم فیلتر Sobel لبها را در هر دو جهت X و Y شناسایی می‌کند و در این تصویر لبها به وضوح و مشخص در نواحی مختلف مثل صورت، لباس، دست‌ها و کلاه شناسایی شده است و این فیلتر برای تشخیص تغییرات تدریجی در شدت روشناکی استفاده می‌شود. فیلتر Prewitt نیز مشابه فیلتر Sobel است اما حساسیت کمتری نسبت به تغییرات جزئی دارد. این فیلتر لبها را در جهت‌های افقی و عمودی شناسایی می‌کند و خروجی آن مشابه Sobel دیده می‌شود اما می‌تواند در مقایسه با Sobel کمی جزئیات کمتری را نشان دهد (لبهای شناسایی شده با استفاده از Sobel معمولاً به‌طور واضح‌تر و با جزئیات بیشتر نسبت به دیگر فیلترها نمایش داده می‌شوند).

فیلتر Laplacian هم برای شناسایی لبها تیز و تغییرات سریع شدت روشنایی در تصویر استفاده می‌شود و همانطور که در تصویر هم می‌بینیم نتیجه اعمال این فیلتر شبیه به یک تصویر با کنترast بالا است که بیشتر تغییرات جزئی را شناسایی کرده است؛ همچنین لبها را شناسایی شده با Laplacian دقیق‌تر و تیزتر از دیگر فیلترها هستند اما از آن طرف ممکن است بعضی از جزئیات نرم‌تر (تغییرات ملائم و تدریجی‌تر) تصویر را از دست بدهد.

برای تصویر نقاشی پنگوئن هم به تبع مثل تصویر پرتره می‌توانیم به همین صورت تجزیه و تحلیل‌ها را انجام دهیم (تصویر خروجی Laplacian برای پنگوئن را اگر زوم را بیشتر کنیم لبها به درستی دیده خواهند شد).

## C

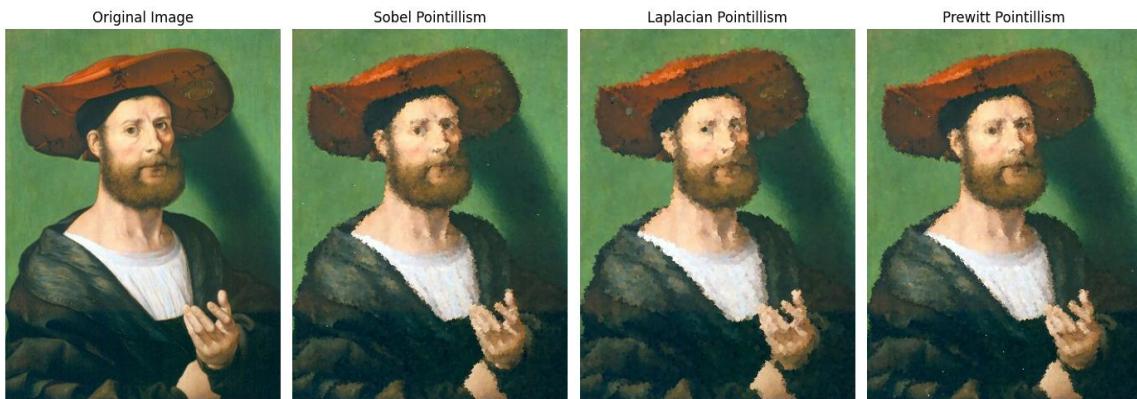
در این بخش باید افکت پوینتیلیسم را پیاده‌سازی کنیم و ابتدا یک تصویر سفید با اندازه مشابه تصویر اصلی ایجاد می‌کنیم و سپس نقاط تصادف روی آن قرار می‌دهیم. در هر نقطه نیز دایره‌ای با شعاع مناسب با موقعیت آن رسم کرده و آن را با رنگ پیکسل متناظر در تصویر اصلی پر می‌کنیم و در نهایت، تصویر پوینتیلیسم را روی دو

تصویر ورودی اعمال می‌کنیم و این موارد را به درستی در کد پیاده‌سازی کردیم و خروجی نهایی تصاویر ورودی را نیز در پارت بعدی نمایش می‌دهیم و به تجزیه و تحلیل آن می‌پردازیم.

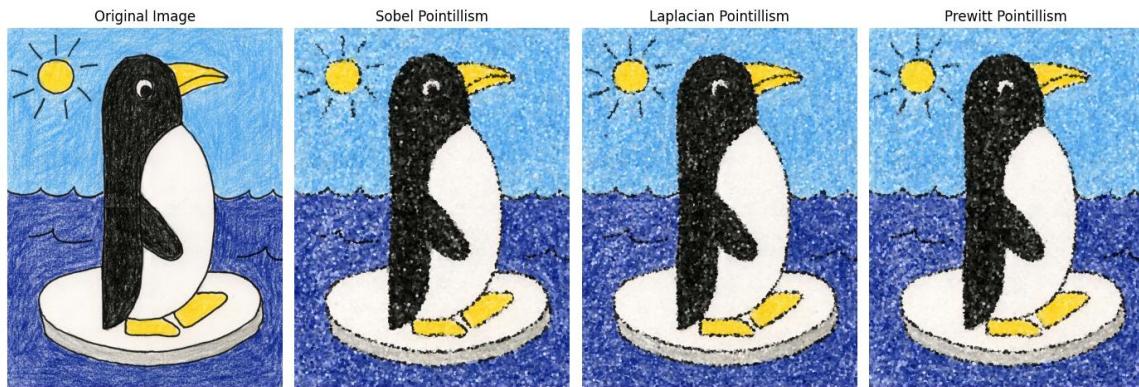
## D

در این بخش اثر پوینتیلیسم را با استفاده از سه روش مختلف تشخیص لبه که در پارت‌های قبلی اشاره کردیم روی دو تصویر اعمال کردیم و همینطور پارامترهایی که داشتیم مثل تعداد نقاط تصادفی، شعاع لبه‌ها و شعاع مناطق غیر لبه‌ای تصاویر را که در پارت ۶ با موارد مختلف امتحان کردیم، بهترین مقداردهی به پارامترها را برای ایجاد تصاویر پوینتیلیسم رو ورودی‌ها را انتخاب کردیم، در این بخش خروجی را نمایش دادیم (اگر به خروجی قسمت‌های مختلف این سه فیلتر به عنوان مثال چشم‌ها یا ... را نگاه کنیم تفاوت‌هایی که این سه فیلتر روی تصاویر ورودی ایجاد کرده‌اند را می‌توانیم مشاهده کنیم):

Pointillism - Image 1



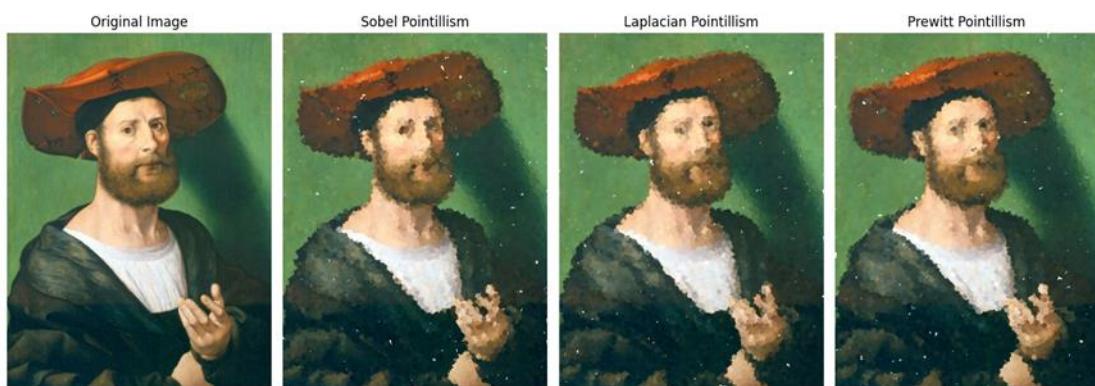
### Pointillism - Image 2



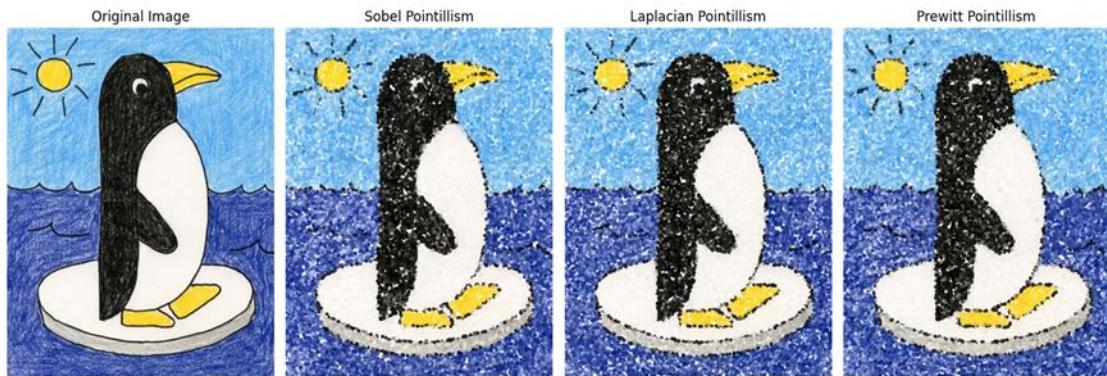
### E

در این پارت هم پارامترها را با مقادیر مختلف مقداردهی کردیم و بهترین ترکیب مقادیر این پارامترها را در نظر گرفتیم و خروجی را در پارت قبل نمایش دادیم. تصویر پنگوئن چون به صورت دستی نقاشی شده است جزئیات بیشتری مثل تصویر پرتره نداشته و به همین تصویر نهایی پوینتیلیسم آن هم به صورت های قبلی ایجاد شد و اگر مثل پرتره کیفیت و جزئیات بیشتری داشت به خوبی می توانست به پوینتیلیسم تبدیل شود اما با این وجود تا جای امکان توانستیم به افکت پوینتیلیسم خواسته شده خروجی را بررسانیم.

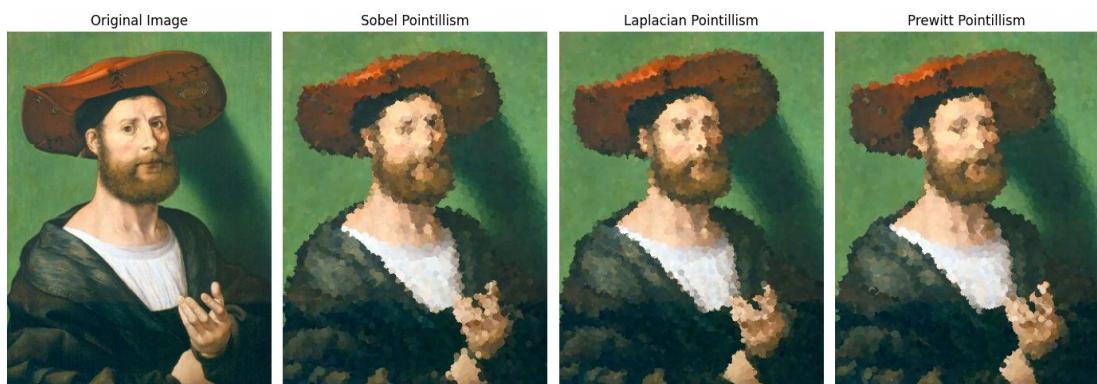
### Pointillism - Image 1



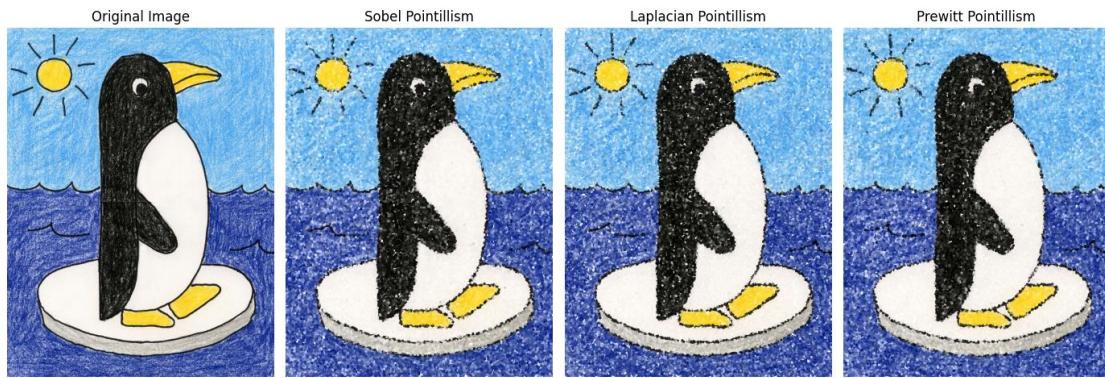
Pointillism - Image 2



Pointillism - Image 1



## Pointillism - Image 2

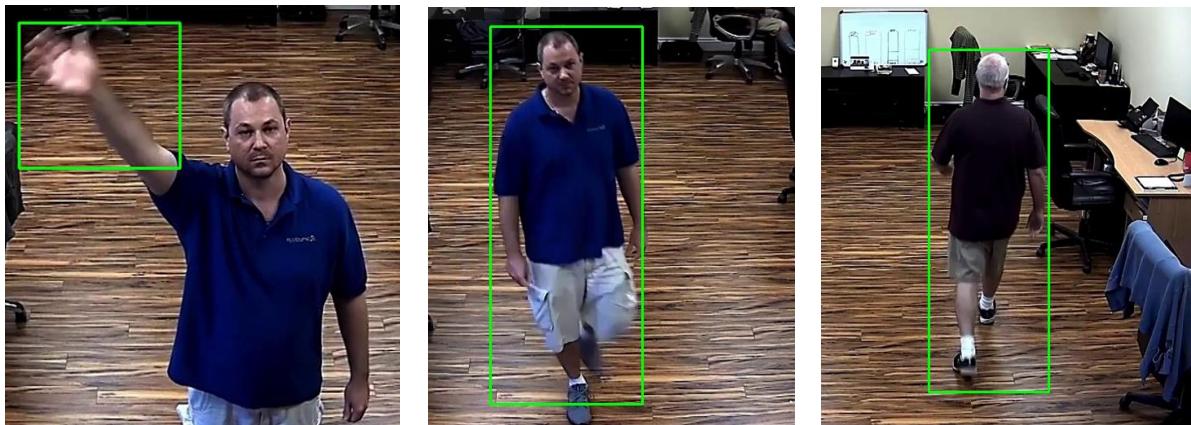


## Problem 4: Automated video surveillance

در این تمرین باید به توسعه یک سیستم تشخیص حرکت با استفاده از تفاوت فریم‌ها پردازیم و هدف اصلی نیز شناسایی حرکت‌های مهم و فیلتر کردن تغییرات ناچیز است تا فقط فعالیت‌های معنادار و مهم ثبت شوند. در این فرایند طبق گفته‌های متن تمرین از تکنیک‌های کاهش نویز مثل Gaussian blur و شناسایی کانتور برای برجسته کردن اشیای متحرک استفاده کردیم.

برای پیاده‌سازی این تمرین طبق روال ابتدا کتابخانه‌های مورد نیاز را import کردیم و پس از اینکه ویدیو را آپلود کردیم، ویژگی‌های آن مانند ابعاد فریم و نرخ فریم استخراج کردیم و سپس یک نویسنده ویدیو برای ذخیره خروجی ایجاد کردیم. برای تشخیص حرکت، از یک روش Subtraction پس زمینه (Background Subtraction) استفاده کردیم که تغییرات موجود در فریم‌ها را شناسایی کند. پس از اعمال بلور Gaussian Dilatation و Erosion جهت کاهش نویز تصویر، تغییرات به کمک آستانه‌گذاری و عملیات مورفولوژیک مثل تقویت شدند تا نواحی متحرک به طور دقیق‌تری شناسایی شوند. در مرحله بعد با استفاده از تکنیک‌های استخراج کانتور، نواحی تغییر یافته را شناسایی کردیم و در صورت تشخیص حرکت قابل توجه، کادر محدود کننده (Bounding Box) اطراف آن نواحی ترسیم می‌شود. در نهایت فریم‌های پردازش شده با کادرهای شناسایی شده در فایل ویدیویی ذخیره می‌شوند.

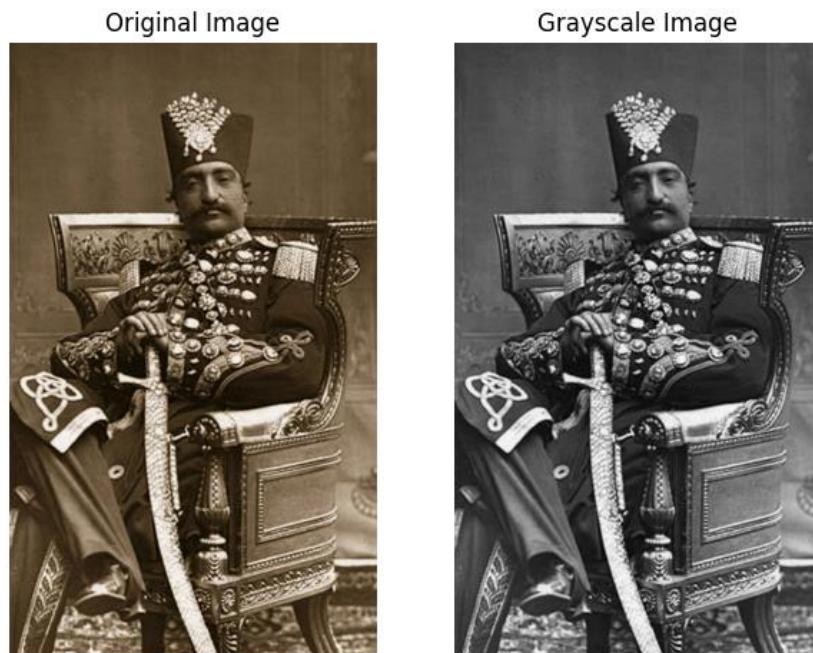
در نهایت هم خروجی ویدیو را می‌توانیم ببینیم که در اینجا چند اسکرین از آن را آوردم:



## Problem 5: Image Denoising Using Spatial Filters

.A

در این تمرین قصد داریم که فیلترهای مکانی را روی تصاویری که با انواع مختلف نویز دستکاری شدن اعمال کنیم تا نویزها را کاهش دهیم و تاثیر اندازه‌های مختلف کرنل را با استفاده از معیار PSNR کنیم. در ابتدا کتابخانه‌های مورد نظر را import می‌کنیم و سپس مثل روال گذشته تصویر اصلی و تبدیل آن به فرم Grayscale را در خروجی نمایش می‌دهیم:



حالا در ادامه باید دو نویز گوسی و نمک-فلفل را به تصویر قبلی اضافه کنیم و در پارت‌های بعد تلاش می‌کنیم که با فیلترهای مکانی مختلف این نویزها را حذف کنیم (به طور کلی نمی‌توانیم با اطمینان راجب حذف نویز صحبت کنیم اما کاهش نویز عبارت بهتر و درست‌تری است). نویز گوسی را با توزیع نرمال (با مقادیر میانگین 0 و واریانس 0.01) به تصویر اولیه اضافه کردیم و بعد مقادیر را در بازه 0 تا 255 نرمال کردیم و در نهایت خروجی نویز گوسی به صورت زیر در آمده است:

Gaussian Noisy Image



نویز نمک-فلفل به صورت تصادفی برخی از پیکسل‌ها را به رنگ مشکی و یا سفید تبدیل می‌کند. این نویز را با احتمال ۵ روی تصویر اصلی اعمال می‌کنیم و یعنی از بین همه پیکسل‌های تصویر به طور میانگین ۵ درصدشون نویزی می‌شوند و البته می‌توانستیم این احتمال را مقادیر دیگری هم بدیم اما نویز خیلی کم یا زیاد نمی‌گذارد تحلیل بهتری از عملکرد و موارد خواسته شده در سوال داشته باشیم. همانطور که گفتیم این نویز باعث ایجاد نقاط تصادفی سیاه (0) و سفید(255) در تصویر می‌شود که شبیه به دانه‌های فلفل و نمک دیده می‌شوند و خروجی این نویز به صورت زیر در آمده است:

Salt and Pepper Noisy Image



## .B

بعد از اینکه نویزهای گفته شده را در تصویر اعمال کردیم و در این بخش باید فیلترهای میانگین، گوسی، میانه و دو طرفه را روی تصاویر نویزی اعمال کنیم و تلاش کنیم که نویز آنها را کاهش دهیم. در ابتدا یک تابع تعریف کردیم که این چهار فیلتر را با اندازه کرنل دلخواه (که در اینجا ما از اندازه  $5 \times 5$  برای آنها استفاده کردیم) روی تصاویر نویزی مرحله قبل اعمال کردیم و برای ارزیابی بهتر مقادیر PSNR هر خروجی را نیز محاسبه می کنیم و در نهایت خروجی های اعمال شده روی هر دو تصویر نویزی را در خروجی نمایش می دهیم:

Mean Filter (Gaussian Noise)  
PSNR: 27.95 dB



Gaussian Filter (Gaussian Noise)  
PSNR: 29.49 dB



Median Filter (Gaussian Noise)  
PSNR: 27.92 dB



Bilateral Filter (Gaussian Noise)  
PSNR: 29.28 dB



Mean Filter (Salt & Pepper Noise)  
PSNR: 24.72 dB



Gaussian Filter (Salt & Pepper Noise)  
PSNR: 24.49 dB



Median Filter (Salt & Pepper Noise)  
PSNR: 30.54 dB



Bilateral Filter (Salt & Pepper Noise)  
PSNR: 21.16 dB



اگر به تصاویر بالا دقت کنیم می بینیم برای تصویر نویزی گوسی ، فیلتر گوسی بهتر از بقیه عمل کرد و بالاتری با مقدار 29.49 نسبت به 3 فیلتر دیگر داشت و برای تصویر نویزی نمک-فلفل نیز فیلتر میانه عملکرد بهتری داشت و با PSNR برابر 30.54 عملکردی خیلی بهتری از سه فیلتر دیگر داشت.

## .C

حالا در این بخش باید اندازه‌های مختلف کرنل را اعمال کنیم و تاثیر آنها روی عملکرد کاهش نویز را بررسی کنیم. برای سایز کرنل ما مقادیر ۳، ۵، ۷ و ۹ را در نظر گرفتیم و بعد با این اندازه‌های مختلف چهار نوع فیلتر مکانی گفته شده را روی تصاویر نویزی اعمال کردیم و همچنین برای ترجیه و تحلیل بهتر مقادیر PSNR را برای هر کدام نیز محاسبه کردیم و در نهایت نیز اعمال این چهار فیلتر روی تصویر نویزی گوسی با اندازه‌های کرنل مختلف را در زیر نمایش دادیم:

Mean (Kernel 3x3)  
PSNR: 28.84 dB



Gaussian (Kernel 3x3)  
PSNR: 28.26 dB



Median (Kernel 3x3)  
PSNR: 27.08 dB



Bilateral (Kernel 3x3)  
PSNR: 26.83 dB



Mean (Kernel 5x5)  
PSNR: 27.95 dB



Gaussian (Kernel 5x5)  
PSNR: 29.49 dB



Median (Kernel 5x5)  
PSNR: 27.94 dB



Bilateral (Kernel 5x5)  
PSNR: 29.27 dB



Mean (Kernel 7x7)  
PSNR: 25.08 dB



Gaussian (Kernel 7x7)  
PSNR: 28.64 dB



Median (Kernel 7x7)  
PSNR: 25.83 dB



Bilateral (Kernel 7x7)  
PSNR: 28.49 dB



Mean (Kernel 9x9)  
PSNR: 23.03 dB



Gaussian (Kernel 9x9)  
PSNR: 27.36 dB



Median (Kernel 9x9)  
PSNR: 23.82 dB



Bilateral (Kernel 9x9)  
PSNR: 26.90 dB



همانطور که در تصویر بالا هم می‌بینیم برای فیلتر میانگین هرچه اندازه کرنل افزایش پیدا کند مقدار PSNR آن کاهش یافته و کمتر توانستیم نویز را کاهش دهیم. برای فیلتر مکانی گوسی وقتی سایز ۳ را به ۵ افزایش دادیم عملکرد بهتر شد اما بعد از آن روند روند نزولی را پیش گرفت و کاهش یافت. برای فیلتر میانه نیز مثل فیلتر گوسی

ابتدا بعد از تغییر اندازه کرنل از 3 به 5 مقدار PSNR افزایش یافت اما بعدش روند نزولی به خود گرفت. برای فیلتر دوطرفه هم به همین صورت است و می‌توان آن را تجزیه و تحلیل کرد.

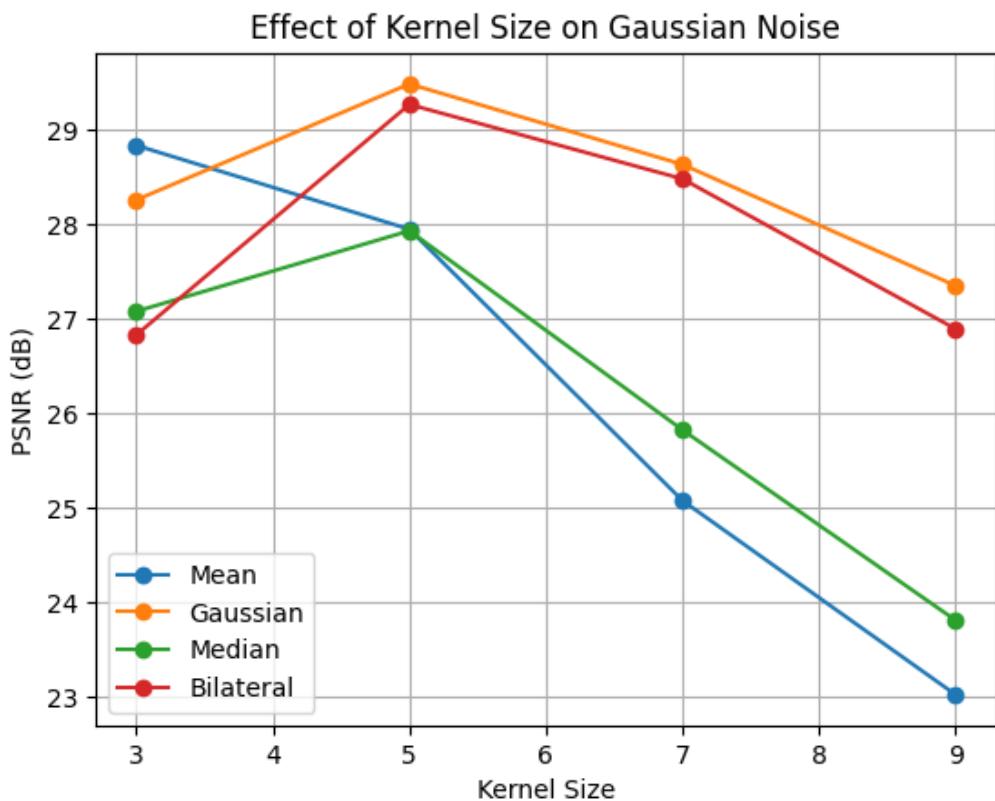
در کرنل با اندازه  $3 \times 3$  فیلتر میانگین با مقدار PSNR برابر 28.84 بهترین عملکرد را داشت.

در کرنل با اندازه  $5 \times 5$  فیلتر گوسی با مقدار PSNR برابر 29.49 بهترین عملکرد را داشت.

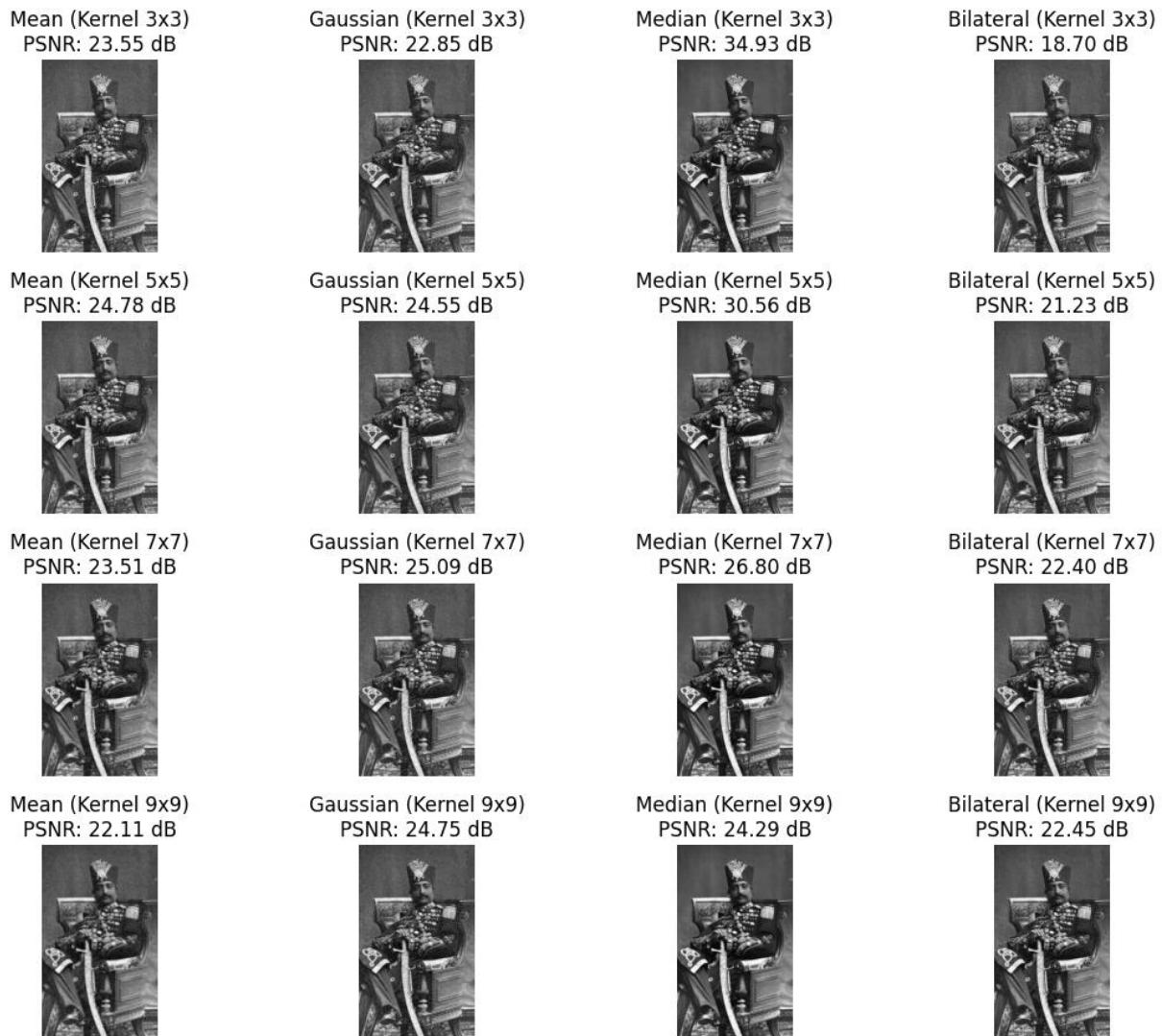
در کرنل با اندازه  $7 \times 7$  فیلتر گوسی با مقدار PSNR برابر 28.64 بهترین عملکرد را داشت.

در کرنل با اندازه  $9 \times 9$  فیلتر گوسی با مقدار PSNR برابر 27.36 بهترین عملکرد را داشت.

تحلیل‌هایی که از روی تصاویر بالا برای تصویر نویزی گوسی انجام دادیم و می‌توانیم از روی نمودار زیر هم مشاهده کنیم و رابطه بین مقدار PSNR و اندازه کرنل را به خوبی می‌توانیم در خروجی زیر ببینیم:



حالا در این قسمت خروجی‌های بالا روی تصویر نویزی نمک-فلفل را می‌خواهیم نمایش دهیم و تحلیل کنیم:



همانطور که در تصویر بالا هم می‌بینیم برای فیلتر میانگین گوسی وقتی سایز 3 را به 5 افزایش دادیم عملکرد بهتر شد اما بعد از آن روند نزولی را پیش گرفت و کاهش یافت. برای فیلتر مکانی گوسی می‌بینیم با افزایش سایز کرنل به 5 مقدار PSNR افزایش یافته و همچنان با افزایش اندازه سیر صعودی گرفت اما وقتی اندازه به 9 تغییر کرد شروع به کاهش کرد. برای فیلتر میانه هر بار که اندازه کرنل را افزایش می‌دهیم مقدار PSNR کاهش پیدا می‌کند و این مقدار با مقدار اندازه رابطه عکس دارد و نمودار آن نیز نزولی اکید می‌باشد. برای فیلتر دوطرفه هم برعکس فیلتر میانه با افزایش اندازه کرنل می‌بینیم که عملکرد این فیلتر در کاهش نویز بهتر می‌شود و روند صعودی را پیش گرفته است.

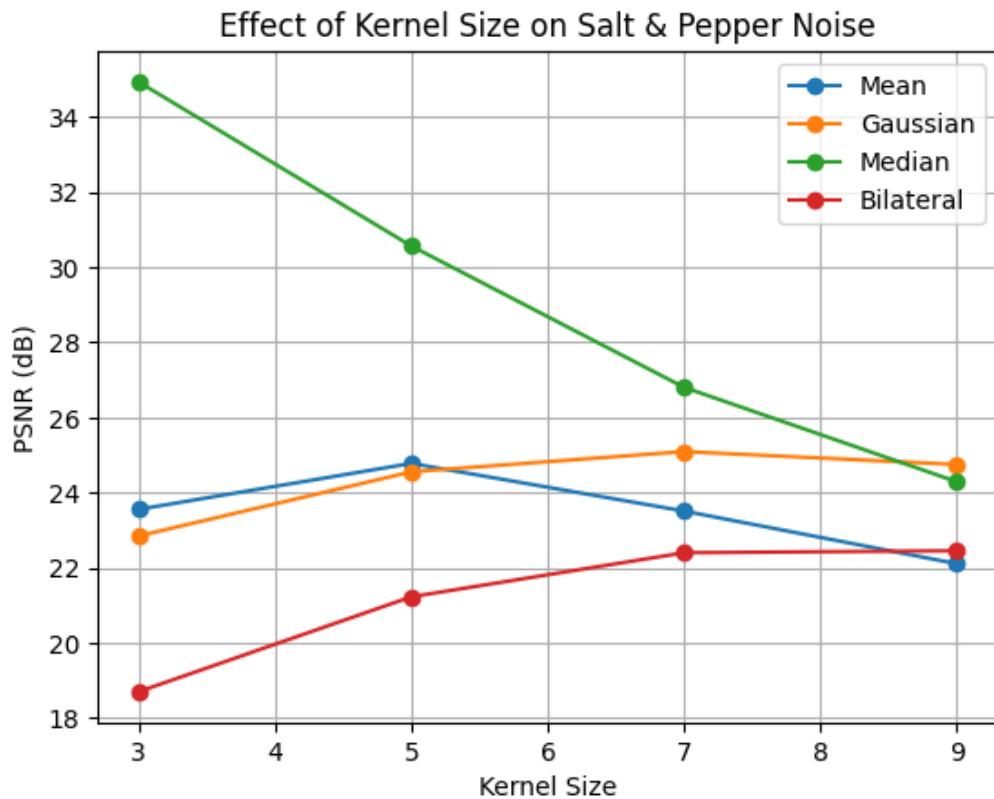
در کرنل با اندازه 3x3 فیلتر میانه با مقدار PSNR برابر 34.93 بهترین عملکرد را داشت.

در کرنل با اندازه 5x5 فیلتر میانه با مقدار PSNR برابر 30.56 بهترین عملکرد را داشت.

در کرنل با اندازه  $7 \times 7$  فیلتر میانه با مقدار PSNR برابر 26.80 بهترین عملکرد را داشت.

در کرنل با اندازه  $9 \times 9$  فیلتر گوسی با مقدار PSNR برابر 24.75 بهترین عملکرد را داشت.

تحلیل‌هایی که از روی تصاویر بالا برای تصویر نویزی گوسی انجام دادیم و می‌توانیم از روی نمودار زیر هم مشاهده کنیم و رابطه بین مقدار PSNR و اندازه کرنل را به خوبی می‌توانیم در خروجی زیر ببینیم:



## .D

این پارت را در بخش قبلی کامل برای هر کدام از فیلترهای مختلف نشان دادیم و هم تصاویر و هم نمودارهای مربوطه را تجزیه و تحلیل کردیم.

## .E

سوال این بخش را نیز در بخش‌های قبلی توضیح دادیم و برای تصویر نویزی گوسی، فیلتر مکانی گوسی با اندازه کرنل  $5 \times 5$  بهترین عملکرد را داشت و با PSNR برابر 29.49 نسبت به فیلترهای دیگر بهتر عمل کرد. برای تصویر نویزی نمک-فلفل هم فیلتر مکانی با اندازه کرنل  $3 \times 3$  میانه بهترین عملکرد را و با PSNR برابر 34.93 نسبت به سایر فیلترها داشت و کاهش نویز بهتری را از خود نشان داد و اگر به نمودارهای پارت C هم مجدد نگاه کنیم می‌توانیم به خوبی این موضوع را ببینیم و در ادامه نیز دو فیلتری که بهترین عملکر را داشتند مجدد نمایش می‌دهیم:

Median (Kernel 3x3)      Gaussian (Kernel 5x5)  
PSNR: 34.93 dB                  PSNR: 29.49 dB



## .F

این بخش را نیز به طور کامل و دقیق هم از روی تصاویر خروجی هر فیلتر، مقدار PSNR و نمودارهای رسم شده در پارت C تجزیه و تحلیل کردیم و برای پرهیز از تکرار مجدد و طولانی شدن گزارش از آوردن همان تجزیه و تحلیل‌ها در این بخش اجتناب کردیم.

## .G

در پردازش تصویر همیشه یک چالش بین کاهش نویز (Noise Reduction) و حفظ لبه‌ها (Edge Preservation) وجود دارد؛ یعنی وقتی می‌خواهیم نویز را از بین ببریم ممکن است جزئیات و لبه‌های مهم تصویر هم از بین بروند. در این بخش باید ترید آف بین کاهش نویز و حفظ لبه‌ها را بررسی کنیم و هر فیلتر یکسری مزایا و معایب دارد که باعث می‌شود بسته به عملکرد و کاربردشان یک مصالحه‌ای بین کاهش نویز و حفظ لبه‌ها را در نظر بگیریم. در ادامه نیز می‌خواهیم چهار فیلتر مکانی گفته شده را بررسی کنیم:

### :Mean Filter ▪

این فیلتر برای کاهش نویز در هر دو تصویر نویزی (گوسی و نمک-فلفل) عملکرد متوسطی داشت و مقدار PSNR به طور کلی در بازه 23.03 تا 28.84 قرار دارد. در تصویر نویز گوسی عملکرد قابل قبولی دارد اما با افزایش اندازه

کرنل تصویر بیش از حد صاف می‌شود و جزئیات و لبه‌ها نیز از بین می‌روند. اگر تصاویر خروجی را با دقت بررسی کنیم مشاهده می‌کنیم که هرچه سایز کرنل بزرگ‌تر شده (مثلاً 9x9) لبه‌ها محو شده‌اند و تصویر حالتی تار پیدا کرده است. برای نویز نمک-فلفل هم عملکرد نسبتاً ضعیفی دارد، زیرا این نوع نویز به صورت نقطه‌ای و ناگهانی است و فیلتر میانگین نمی‌تواند مقادیر پرت را حذف کند. بنابراین PSNR در این تصاویر هم پایین مانده است.

#### :Gaussian Filter ▪

این فیلتر نسبت به فیلتر میانگین عملکرد بهتری از خود نشان داده است و به دلیل وزن دهنده به پیکسل‌های اطراف نویز را بهتر کاهش می‌دهد. مقدار PSNR بین 22.85 تا 29.49 قرار دارد که در تصویر نویز گوسی و مخصوصاً با کرنل 5x5 بهترین عملکرد حاصل شده است. در حفظ لبه‌ها نیز عملکرد بهتری نسبت به فیلتر میانگین دارد ولی همچنان با افزایش سایز کرنل، لبه‌ها محو می‌شوند. در تصویر نویز نمک-فلفل، فیلتر گوسی عملکرد خوبی ندارد و در کرنل 9x9 به PSNR 24.75 رسیده است که همچنان پایین‌تر از فیلتر میانگین است.

#### :Median Filter ▪

این فیلتر به صورت خاص برای حذف نویز نمک-فلفل بسیار مناسب است. در تصویر مربوط به نویز نمک-فلفل با کرنل سایز 3x3 مقدار PSNR تا 34.93 افزایش یافته که بالاترین مقدار در بین تمام نتایج است. این فیلتر برخلاف فیلترهای دیگر به میانگین وابسته نیست، بلکه مقدار میانه را می‌گیرد و همین باعث می‌شود مقادیر پرت که در نویز نمک-فلفل زیاد هستند حذف شوند. برای نویز گوسی اما عملکرد چندان خوبی ندارد، چون این نوع نویز پراکنده‌تر است و فیلتر میانه نمی‌تواند آن را به خوبی حذف کند؛ همچنان با افزایش سایز کرنل نیز PSNR کاهش می‌یابد و لبه‌ها نیز دچار شکستگی می‌شوند. به طور کلی برای نویز نمک-فلفل بهترین عملکرد را دارد ولی برای نویز گوسی مناسب نیست.

#### :Bilateral Filter ▪

این فیلتر یکی از پیچیده‌ترین فیلترها از نظر محاسباتی است ولی ترید آف خوبی بین کاهش نویز و حفظ لبه‌ها ایجاد می‌کند. در تصویر نویز گوسی عملکرد خوبی داشته و با کرنل سایز 5 مقدار PSNR به 29.27 رسیده است. مزیت بزرگ آن این است که برخلاف فیلترهای میانگین و گوسی می‌تواند لبه‌ها را حفظ کند، چون در فیلتر کردن هم فاصله فضایی و هم تفاوت شدت پیکسل‌ها را در نظر می‌گیرد. اما در تصویر نویز نمک-فلفل عملکرد مناسبی ندارد و حتی در کرنل 3x3 مقدار PSNR به 18.70 کاهش یافته است. با افزایش کرنل مقدار PSNR کمی بهتر شده ولی همچنان نسبت به فیلتر میانه بسیار ضعیفتر است و دلیل این موضوع این است که فیلتر دوطرفه برای نویزهای ناگهانی و شدید مثل نمک-فلفل طراحی نشده و بیشتر برای نویزهای نرم‌تر مثل نویز گوسی مناسب است.

## Problem 6: Hands-On Image Processing: Exploring Visual Computation

.A

در این سوال بدین معنی پردازیم که چرا تابع  $y_{\text{loss}}$  یکسانسازی لسته به طور کم همیشه گرام مصلح تغییر نماید.

$$S = T(r) = (L-1) \int_0^r Pr(w) dw$$

ضرمول سوسته:

$$S_K = T(r_K) = \text{round} \left( (L-1) \sum_{j=0}^K Pr(r_j) \right)$$

$$= \text{round} \left( (L-1) \sum_{j=0}^K \frac{n_j}{n} \right)$$

تایید یکسانسازی همیشه گرام بکار یکنیز بجایی کنتر است در پردازش تعمیر است که با بازنگری مقادیر سدت روشنایی یکسانها همیشه گرام مصلح و تغییر نماید.

راشد ضرولت یکنواخت تر و نزدیک حالت مصلح و تغییر تبدیل نماید.

همین باتر ضرولت های مفهومی لسته و سوسته را هم دیگر و حال بدهیم و خوب است:

(I) تایید تبدیل کردن در حالت لسته: نایج  $l = 20$  باعث تبدیل سدت مقادیر بدنزدیت ترین عدد صحیحی سود و همین موضوع می تواند سبب شود که جندین مقدار مختلف خروجی باشد (مقدار خروجی تبدیل سوند (۱۰۹۷) یا ۲۰۵۱) به خروجی ۲ (نیاست می سوند) درنتیج تعداد سطح سدت سی از یکسانسازی کاهش می یابد و باعث کاهش یکنواختی همیشه گرام می سود و

(II) توزیع نامنفلح همیشه گرام و روایی: اگر همیشه گرام و روایی دارای یکدیگر همان پسیار پریزگ یا کوچک باشد ممکن است باز همیان تایید تبدیل سدت نشود.

۰۷) اگر تابع توزیع تکه‌ایی هر اول را داده باشد  $CDF$  یک تابع پل‌الاتخانی است و در نتیجه ممکن است در بعضی بازه‌ها تغیرات نداشته باشد (در سطوح ۰۸) سدّت نداشته باشد و سطوح سدّت خروجی در این بازه‌ها تغییر کنند و یا در بخش‌هایی از  $CDF$  متعددی است به صورت ناتوانانی جمیع ۰۹) داشته باشیم و تغیرات ناتوانانی بدیگران سازی هیستوگرام آسیب می‌زنند و باعث می‌شود هیستوگرام بدل مسطح ایجاد شود.

۱۰) خلاصه انتیزاسیون: در تبدیل پیوسته به لسته خلاصه انتیزاسیون می‌تواند منجر به تلفیق سطوح سدّت سود و می‌تواند روی مسطح سدّت یا ۱۱) سدن اگر بلند باشد،

۱۲) محدودیت تعداد پیکسل همانند تعداد پیکسل ها در یک تغییر مسیرها با تعداد سطوح سدّت ممکن است که در خروجی یکسان سازی فراهم ۱۳) ارتباط دارد و می‌تواند باعث عدم تغییر یک هیستوگرام کاملاً مسطح شود. همچنین محدودیت تعداد پیکسل می‌تواند موجب کمبود تنوع سدّت ۱۴) در تغییر شود و فقط تعداد محدودی از سطوح سدّت فعلی شود و ۱۵) همچنین این مورد باعث پراکندگی بعدن سدّت ها هم می‌تواند شود و در نتیجه هیستوگرام نتواند کاملاً مسطح شود.

B

07 از درس فرمول ها کنترل برای داریم

$$1) S = T(r) = \int_0^r P_r(w) dw \quad 08$$

$$2) G(z) = \int_0^z P_z(t) dt = S \rightarrow G(z) = S \rightarrow \quad 09$$

$$3) z = G^{-1}(S) = G^{-1}[T(r)] \quad 10$$

حالا اگر دو پرسه داریم که این سوال را بررسی کنیم می بینیم محور افقی که در

محوار دارد سرتیپ می باشند و زیرا اسما نیز این محور عبور

11 هم مقادار چنانچه احتمال یافتن  $P_r(r)$  و  $P_z(z)$  می باشد بنودار را می بینیم

محوری و چیزی نزولی است و می توانیم با این صورت بنویسیم که:

$$P_r(r) = 2(1-r) \quad 12 \quad 0 < r < 1$$

$$P_z(z) = 2z \quad 0 < z < 1 \quad 13$$

حالا اگر فرمول اول یعنی  $S = T(r) = \int_0^r P_r(w) dw$  با جایگذاری

14 محل مقادار  $S$  را بدست می آوریم:

$$S = T(r) = \int_0^r 2(1-w) dw = 2w - w^2 \Big|_0^r = 2r - r^2 \quad 15$$

حالا از فرمول دوچه برای  $G(z)$  استفاده می کنیم:

$$G(z) = \int_0^z 2t dt = t^2 \Big|_0^z = z^2 \quad 16$$

$$G(z) = S = z^2 \rightarrow z = \sqrt{S} \quad 17$$

حالا از فرمول سوچی که  $z = G^{-1}(S)$  است می بینیم

$$z = G^{-1}(S) = \sqrt{S} \quad 18$$

است و داریم

1402

07 از طرفی معان اول نتیجه که  $T(r)$  است و کافی است که

مقادار  $T(r)$  را زیر رادیکال قرار دهیم تا مساوی بر حسب  $S$  کو

$$z = G^{-1}(S) = G^{-1}[T(r)] \quad 08$$

حاصل شد:

$$\sqrt{S} = \sqrt{T(r)} = \sqrt{2r - r^2} \quad 09$$

$$\boxed{z = \sqrt{2r - r^2}} \quad 10$$

## C.

در سؤال لفته سده که  $w$  حاصل ضرب  $w_1$  و  $w_2$  است و  
یک کریل جدا از دیگر را می توان به صورت حاصل ضرب خارجی دو بودار

$$w = \begin{bmatrix} 2 & 1 & 1 & 3 \\ 4 & 2 & 2 & 6 \\ 2 & 1 & 1 & 3 \end{bmatrix}, w = w_1 * w_2$$

برای کریل جدا از دیگر باید تمام سطرها یا ستون های کریل ضرب خارجی  
مکله دیگر باشند یا مثلاً  $w$  می بینیم ن سطر دوم ضرب ۲ سطر اول  
ناتمام است و برای ستوان ها هم بدهمین صورت می توانیم بروی لیم  
و مثلاً ستون ۴ و سوم ضرب ۲ ستون اول هستند یا ستون چهارم  
ضرب ۳ ستون اول است سه این کریل همانطور در سؤال لفته

سد جدا از دیگر است.  $w_1$  یک بودار ستوانی  $n \times 1$  و  $w_2$  هم باید  
یک بودار سطی  $1 \times m$  باشد که در نتیجه  $w$  با  $n \times m$  حاصل

$$w_1 = \begin{bmatrix} a \\ b \end{bmatrix} \text{ و } w_2 = \begin{bmatrix} c & d \end{bmatrix}_{1 \times 2} \quad \text{لذو. مثلاً داریم:}$$

$$w = w_1 * w_2 = \begin{bmatrix} a \cdot c & a \cdot d \\ b \cdot c & b \cdot d \end{bmatrix}_{2 \times 2}^{n \times m}$$

حالا اگر ب  $w$  دست کشم می بینم که همه ستون ها ضربی از ستون دوم  
(یا سوم) هستند سه  $w_1$  ما برابر  $\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$  می شود.

+ اما سطرها هم هم سطرها ضربی از سطر اول (یا سوم) هستند

پس  $w_2$  برابر  $\begin{bmatrix} 2 & 1 & 1 & 3 \end{bmatrix}$  می شود و به اک اطمینان داریم:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 3 \\ 4 & 2 & 2 & 6 \\ 2 & 1 & 1 & 3 \end{bmatrix} \Rightarrow w = w_1 * w_2$$

روز درختکاری - روز آموزش همکاری حفظ محیط زیست 1403

30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 09 08 07 06 05 04 03 02 01

## .D

آنودار (a) از مقدار  $A$  شروع شده است و سپس  $r=0$  07  
 در  $r=0$  به سمت  $\infty$  میل کرده است و کم کابع نزولی 08  
 $\rightarrow$  قابع  $e^{-\alpha r^2}$  هم کم کابع نزولی است که از  $1$  شروع شده و اگر آن را در ضرب کنیم داریم 09  
 $\textcircled{I} T(r) = A \cdot e^{-\alpha r^2}$   
 در  $r=0$  مقدار  $A$  می شود و در  $r=\infty$  هم مقدار  $A$  می خواهد 10  
 حالا می توانیم نکته  $L$  را فراخیم تا مقدار  $A$  را هم حساب کنیم 11  
 $T(L_0) = A \cdot e^{-\alpha L_0^2} = \frac{A}{2} \rightarrow e^{-\alpha L_0^2} = \frac{1}{2} \rightarrow$   
 $\alpha = \frac{\ln 2}{L_0^2}$  12  
 آنودار (b) هم کم کابع صعودی است و باید از  $(1 - e^{-\alpha r^2})$  13  
 استفاده کنیم و بعد در ضرب  $B$  ضرب کنیم:  
 $\textcircled{II} T(r) = B \cdot (1 - e^{-\alpha r^2})$   
 در  $r=0$  مقدار  $B$  می شود و در  $r=\infty$  مقدار  $B$  می شود 14  
 آنودار (c) هم صعودی است و باید از  $(1 - e^{-\alpha r^2})$  استفاده 15  
 کردم که  $0$  تا  $1$  رفت و حالا اینجا از  $C$  تا  $D$  می رود و پس اگر  
 در  $(D - C)$  ضرب کنیم از  $0$  تا  $D - C$  می رود و در نهایت اگر 16  
 مقدار  $C$  را هم بدآن اضافه کنیم از  $C$  تا  $D$  می رود: 17  
 $\textcircled{III} T(r) = C + (D - C) (1 - e^{-\alpha r^2})$   
 هر کدام از سه تساوی بالا می توانیم راهنم فراخیم 18  
 $\alpha = \frac{\ln 2}{L_0^2}$