

دانشكده مهندسي برق

**گزارش تمرین دوم درس پردازش تصاویر پزشکی**

**استاد:** جناب آقای دکتر غفاری

**دانشجو:** محمدّصادق کلامی یزدی

**شماره دانشجویی:** 402811068

**تاریح ارسال:** 23/09/1403

پاییز 1403



فهرست مطالب

[فصل 1: تمرین‌های کامپیوتری 4](#_Toc185233691)

[1‏-‏1‏-‏ تبدیل هاف و مورفولوژی 5](#_Toc185233692)

[1‏-‏2‏-‏ تمرین 2 11](#_Toc185233693)

[1‏-‏3‏-‏ ناحیه‌بندی 13](#_Toc185233694)

[1‏-‏4‏-‏ تمرین 4 17](#_Toc185233695)

[فصل 2: تمرین‌های مرتبط با کلاس حل تمرین 18](#_Toc185233696)

[2‏-‏1‏-‏ تمرین 1 19](#_Toc185233697)

1. تمرین‌های کامپیوتری
   1. تبدیل هاف و مورفولوژی

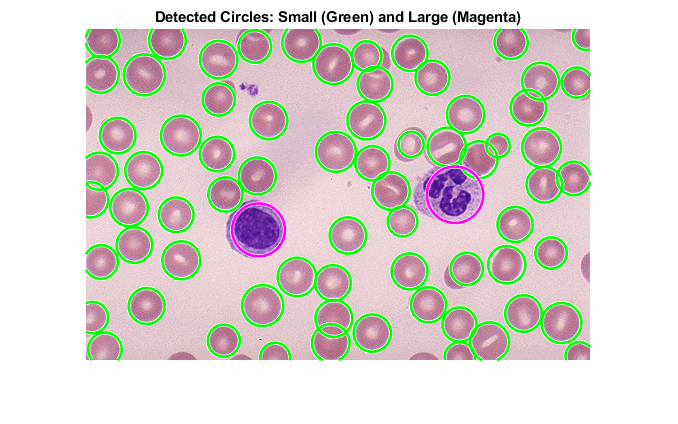
1. به کمک روش تبدیل هاف (Hough Transform) سکه ها و سلولهای زیر را به دو دسته ی سکه ها (سلول‌ها)ی کوچک و سکه ها سلولهای بزرگ تقسیم کنید.

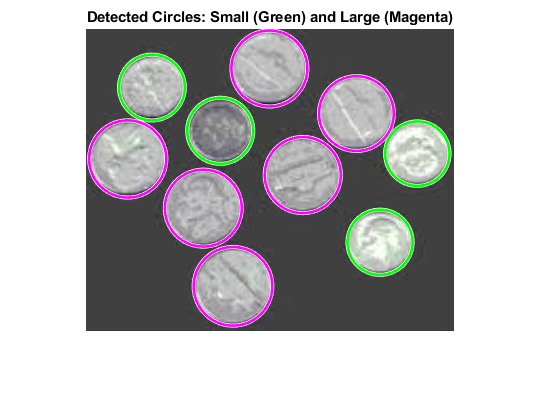
خاکستری کردن، تصویر را ساده می کند و پردازش آن را آسان تر می کند و در عین حال تمام جزئیات مهم مورد نیاز برای تشخیص را حفظ می کند.

در مرحله بعد، تشخیص لبه با استفاده از روش Canny برای برجسته کردن مرزهای اجسام مدور انجام شد. این لبه‌ها به‌عنوان پایه‌ای برای تبدیل دایره هاف عمل می‌کردند، که سپس برای مکان‌یابی دایره‌ها در تصویر با شناسایی مراکز و شعاع‌های آن‌ها استفاده شد.

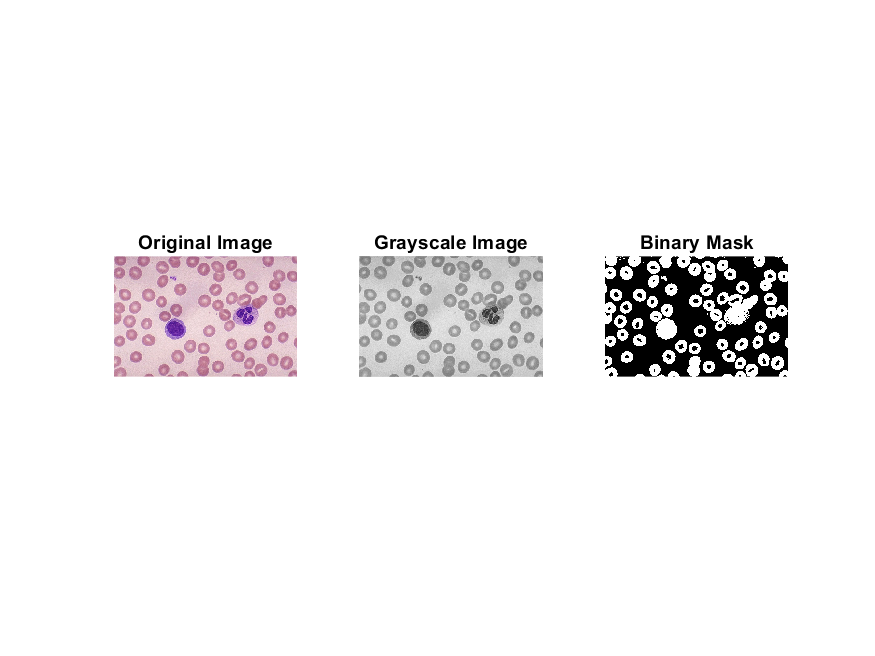
پس از شناسایی دایره ها، بر اساس اندازه آنها به دو دسته تقسیم شدند. دایره هایی با شعاع کوچکتر از آستانه تعیین شده به عنوان "کوچک" طبقه بندی شدند، در حالی که دایره هایی با شعاع بزرگتر به عنوان "بزرگ" برچسب گذاری شدند. سپس دایره‌های شناسایی شده روی تصویر اصلی قرار گرفتند: دایره‌های کوچک با رنگ سبز و دایره‌های بزرگ به رنگ سرخابی برجسته شدند تا بتوان بین این دو گروه تمایز قائل شد.

تصویر نهایی این فرآیند را نشان می‌دهد، با دایره‌های کوچک و بزرگ که به‌دقت شناسایی و دسته‌بندی شده‌اند. این رویکرد کاربردهای عملی در زمینه هایی مانند تصویربرداری پزشکی و مرتب سازی اشیا دارد، جایی که تجزیه و تحلیل تفاوت اندازه ضروری است.



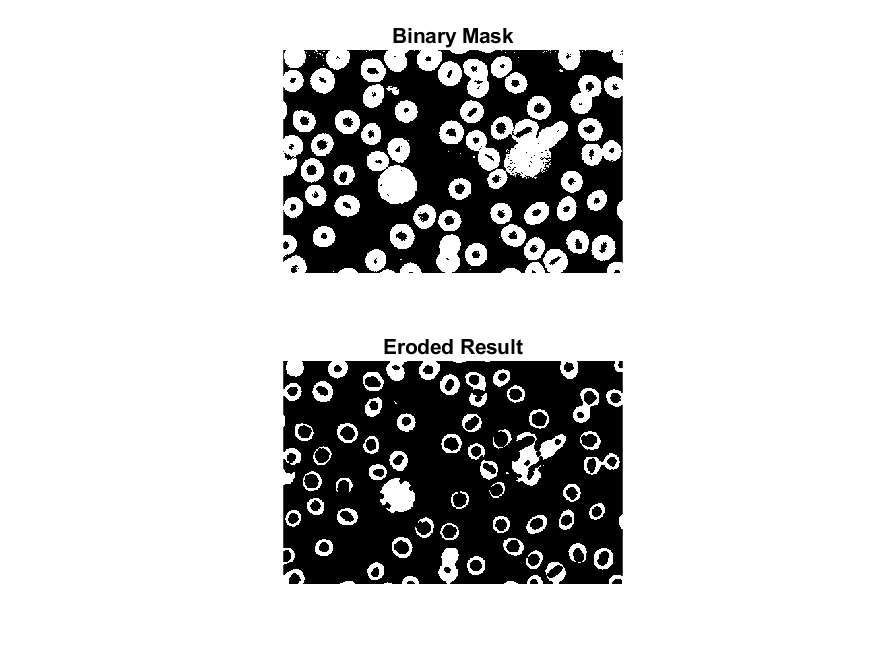
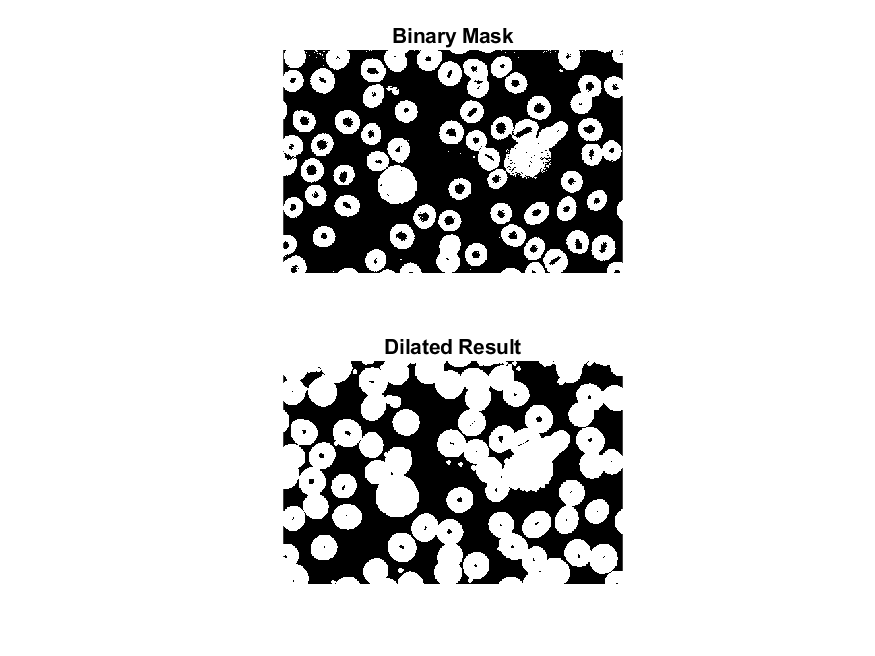


2. روش های پایه مورفولوژی را از جمله opening erosion dilation و closing بر روی تصویر باینری شده سلول خون اجرا نمایید و نتایج خود را تفسیر نمایید.



تصاویر مراحل انجام شده برای تقسیم بندی را نشان می دهند. "تصویر اصلی" ورودی خام را نشان می دهد و ساختارهای سلول را برجسته می کند. سپس، "تصویر مقیاس خاکستری" تصویر اصلی را با تبدیل آن به سایه های خاکستری ساده می کند و پردازش آن را در عین حفظ جزئیات مهم آسان تر می کند.

در نهایت، "Binary Mask" تصویر خاکستری را به یک فرمت سیاه و سفید واضح تبدیل می کند. نواحی سفید نمایانگر سلول‌ها هستند، در حالی که نواحی سیاه پس‌زمینه را مشخص می‌کنند و تمرکز روی سلول‌ها را برای تجزیه و تحلیل یا پردازش بیشتر آسان‌تر می‌کنند.

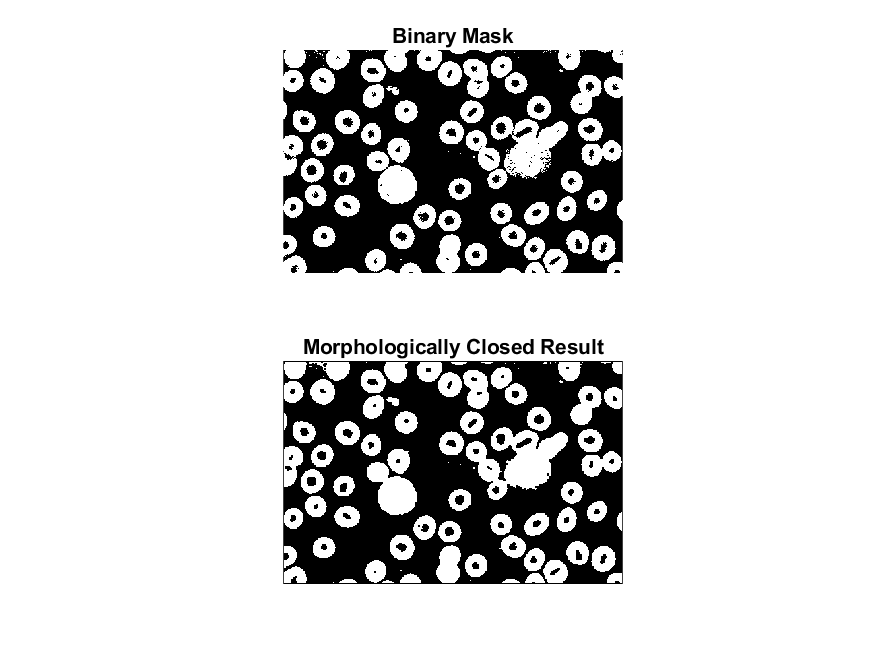
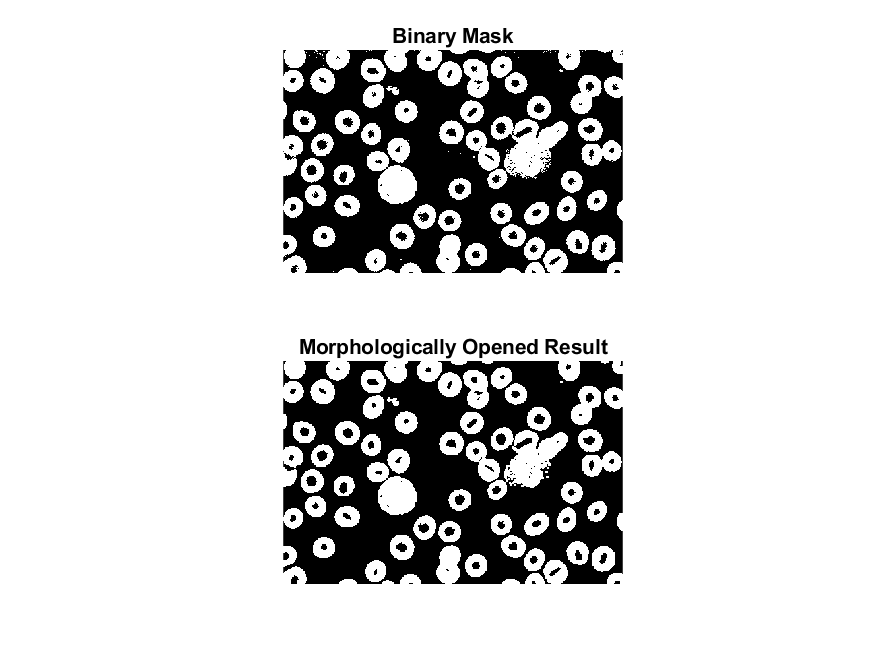


نتیجه تصویر dilation:

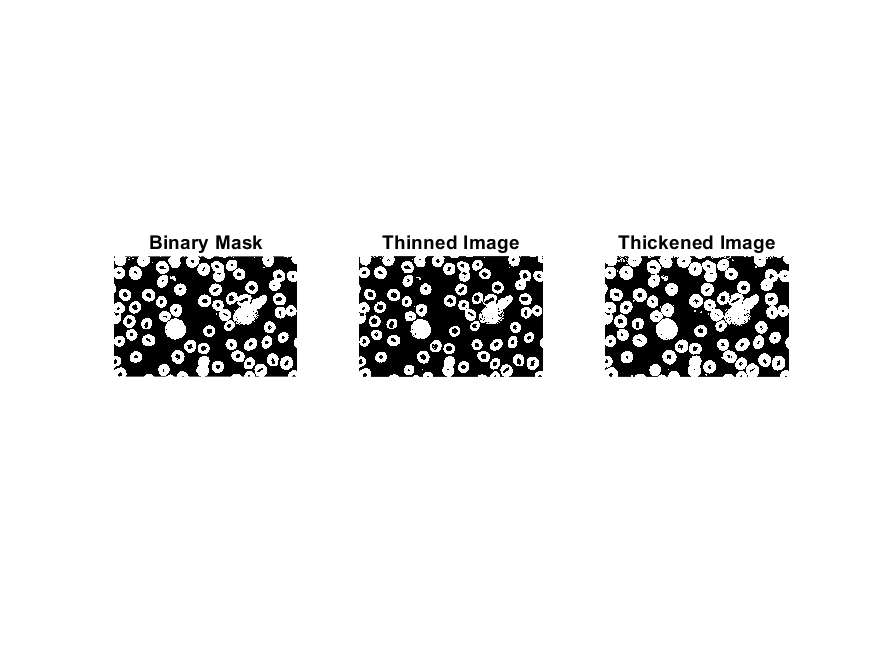
تصویر حاصل از عملیات dilation نشان‌دهنده گسترش مرزهای اشیاء در تصویر باینری است. در پردازش تصویر، dilation یک تبدیل مورفولوژیکی است که مرزهای نواحی سفید (پیش‌زمینه) را گسترش داده و نواحی سیاه (پس‌زمینه) را کاهش می‌دهد. این فرآیند با استفاده از یک structuring element انجام می‌شود که به افزایش اندازه نواحی سفید و اتصال اجزای جدا افتاده کمک می‌کند. عملیات dilation برای پر کردن شکاف‌های کوچک بین اجزا، پر کردن حفره‌های کوچک در داخل اشیاء و تقویت یکپارچگی ساختاری نواحی بخش‌بندی‌شده بسیار مفید است. در نتیجه حاصل، سلول‌ها بزرگ‌تر و به هم متصل‌تر به نظر می‌رسند، که این حالت برای کاربردهایی که به اتصال و تکمیل اشیاء نیاز دارند، مانند آماده‌سازی برای clustering یا صاف کردن مرزها، مناسب است.

نتیجه تصویر Eroded:

Erosion یک عملیات مورفولوژیکی است که برعکس dilation عمل کرده و مرزهای اشیاء را در تصویر باینری کوچک می‌کند. با حذف پیکسل‌ها در لبه‌های نواحی سفید، erosion به جداسازی اجزای متصل، حذف نویز و بهبود مرزهای اشیاء کمک می‌کند. این عملیات به ویژه در جداسازی اشیاء نزدیک به هم و کاهش برجستگی مصنوعات جزئی مؤثر است. در نتیجه حاصل، سلول‌ها کوچک‌تر و متمایزتر دیده می‌شوند، که این ویژگی برای کاربردهایی که به تجزیه و تحلیل دقیق مرزها یا اصلاح بخش‌بندی نیاز دارند، مانند تصویربرداری پزشکی که در آن باید سلول‌های نزدیک به هم تفکیک شوند، مناسب است.

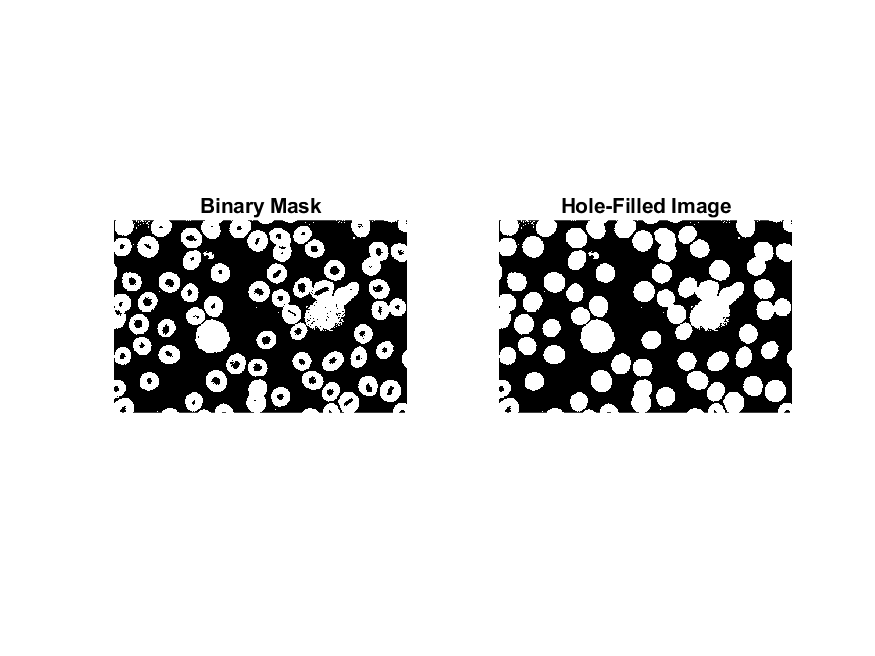


عملیات Opening و Closing دو عملیات پایه‌ای مورفولوژیکی هستند که به طور گسترده در پردازش تصویر برای بهبود تصاویر باینری و ارتقاء ساختار آن‌ها استفاده می‌شوند. Opening ترکیبی از Erosion و سپس Dilation است که با هدف حذف اشیاء کوچک و نویز از پیش‌زمینه بدون تغییر قابل‌توجه در شکل یا اندازه اشیاء بزرگ‌تر به کار می‌رود. این عملیات به ویژه برای جدا کردن اشیاء نزدیک به هم و صاف کردن مرزها از طریق حذف برآمدگی‌ها و نویزهای کوچک مؤثر است. از سوی دیگر، Closing شامل اعمال Dilation و سپس Erosion است که هدف آن پر کردن حفره‌های کوچک، شکاف‌ها یا ناپیوستگی‌ها در پیش‌زمینه بدون تغییر ساختار کلی تصویر می‌باشد. این عملیات برای اتصال اجزاء جدا شده یا ترکیب مناطقی که با شکاف‌های باریک از هم جدا شده‌اند، مانند خطوط نازک یا ترک‌ها، بسیار کارآمد است. این دو عملیات ابزارهای اساسی برای پیش‌پردازش در وظایفی مانند بخش‌بندی، تشخیص اشیاء و تصویربرداری پزشکی محسوب می‌شوند و تضمین می‌کنند که نمایشی مقاوم‌تر و بدون نویز از نواحی موردنظر ارائه شود. در حالی که Opening بر جداسازی اشیاء و کاهش نویز تمرکز دارد، Closing بر روی اتصال و تکمیل اشیاء تمرکز دارد و این تکنیک‌های مکمل برای دستیابی به یک تصویر باینری بهینه ضروری هستند.



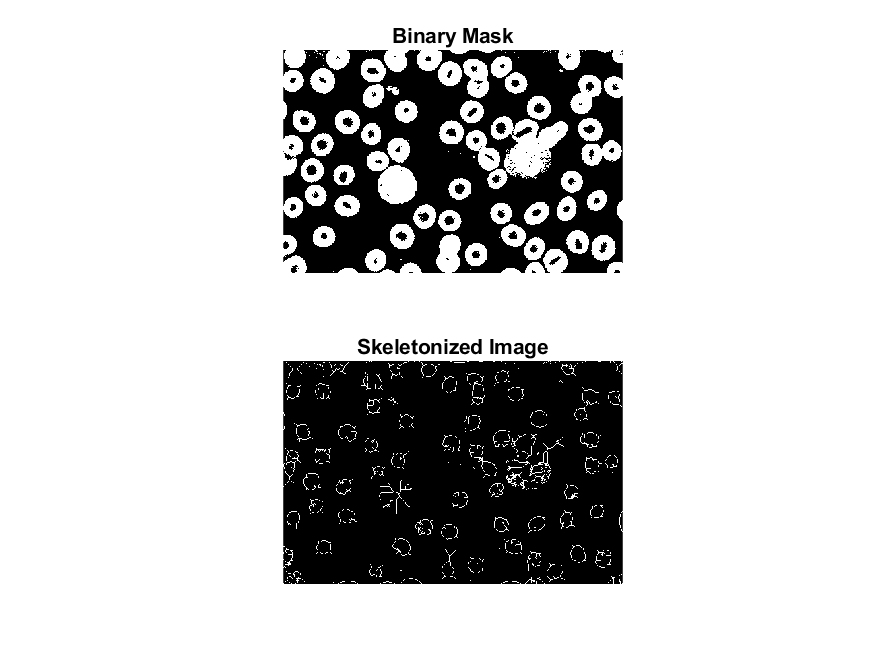
عملیات **Thinning** و **Thickening** دو تکنیک مورفولوژیکی مهم در پردازش تصویر هستند که برای اصلاح و تغییر شکل اجسام در تصاویر باینری استفاده می‌شوند. **Thinning** به منظور نازک‌سازی اجسام با حذف تدریجی لایه‌های پیکسلی از مرزها انجام می‌شود تا ساختار اصلی به یک اسکلت تک‌پیکسلی کاهش یابد. این فرآیند در کاربردهایی مانند استخراج خطوط مرکزی، شناسایی ویژگی‌های ساختاری و تحلیل اشیاء باریک مانند رگ‌ها در تصاویر پزشکی بسیار موثر است.

در مقابل، **Thickening** با افزودن پیکسل‌ها به مرزهای اجسام، به ضخیم‌تر کردن آن‌ها می‌پردازد و برای تقویت و برجسته‌سازی نواحی باریک یا تقویت ساختارهایی که ممکن است در نتیجه نویز یا فرسایش آسیب دیده باشند، کاربرد دارد. این عملیات در ترکیب با سایر تکنیک‌های مورفولوژیکی، ابزاری قدرتمند برای تحلیل شکل و شناسایی دقیق‌تر اشیاء در تصاویر باینری فراهم می‌کند. به طور کلی، **Thinning** بر حفظ ساختار داخلی و نازک‌سازی تمرکز دارد، در حالی که **Thickening** بر تقویت و بهبود مرزها و ساختارهای نازک تمرکز می‌کند.



عملیات **Filling** شامل پر کردن حفره‌ها یا نواحی خالی محصور شده در داخل اشیاء در تصاویر باینری است. این فرآیند را می‌توان به اجزای ساده‌تر مانند **dilation** و سایر تکنیک‌های مورفولوژیکی تقسیم کرد. در واقع، **Filling** با شناسایی نواحی خالی (سیاه) که کاملاً توسط پیش‌زمینه (پیکسل‌های سفید) محصور شده‌اند، شروع می‌شود و سپس با استفاده از تکنیک‌هایی مانند **dilation**، این حفره‌ها را پر می‌کند.

**Dilation** در این فرآیند به تدریج مرزهای نواحی سفید را به سمت داخل گسترش می‌دهد و حفره‌های کوچک را پر می‌کند. این عملیات تا زمانی ادامه می‌یابد که هیچ ناحیه خالی محصور شده‌ای باقی نماند. عملیات **Filling** در کاربردهای متعددی از جمله پزشکی برای تحلیل تصاویر سلولی، پر کردن فضای داخلی سلول‌ها یا حذف شکاف‌های داخلی، بسیار کاربردی است. این تکنیک با بهره‌گیری از **dilation** به عنوان یک ابزار پایه، ساختاری پیوسته‌تر و بدون نقص در پیش‌زمینه ایجاد می‌کند و به تحلیل دقیق‌تر اشیاء کمک می‌کند.



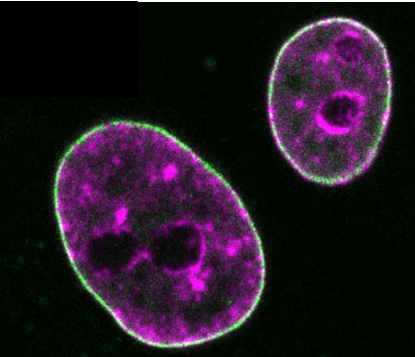
عملیات **Skeletonized** یا اسکلت‌سازی هدف آن کاهش اجسام باینری به ساختارهای مینیمالیستی است که همچنان ویژگی‌های اصلی شکل آن‌ها را حفظ می‌کند. این فرآیند را می‌توان به اجزای ساده‌تری مانند **erosion** و **logical operations** تجزیه کرد. در اصل، **Skeletonized** با استفاده از erosion تدریجی لایه‌های بیرونی جسم را حذف می‌کند و به طور هم‌زمان بررسی می‌کند که ساختار اتصال و ویژگی‌های شکل اصلی حفظ شوند.

**Erosion** لایه‌های پیکسلی از مرز اشیاء را کاهش می‌دهد، در حالی که ساختارهای مرکزی که خطوط اصلی شکل جسم را تشکیل می‌دهند، دست‌نخورده باقی می‌مانند. این عملیات تا زمانی ادامه پیدا می‌کند که اشیاء به خطوط تک‌پیکسلی (اسکلت) کاهش یابند. نتیجه این فرآیند یک اسکلت مرکزی است که ساختار کلی جسم را با حداقل تعداد پیکسل‌ها نشان می‌دهد.

عملیات **Skeletonized** در کاربردهایی مانند تحلیل ساختارهای زیستی، شناسایی مسیرها یا رگ‌های خونی در تصاویر پزشکی، و استخراج ویژگی‌های شکل برای طبقه‌بندی و تحلیل دقیق استفاده می‌شود. این تکنیک با استفاده از مفاهیم ساده‌تر مانند **erosion**، یک نمایش کارآمد و فشرده از ساختارهای پیچیده ارائه می‌دهد.

* 1. تمرین 2

(۲) به کمک روشهای مورفولوژی و یکی از روشهای تشخیص لبه (Edge Detection)، نظیر Canny و یا روش های گفته شده در درس، مرزهای خارجی شکل زیر را بدست آورید.



Extracted Boundaries (Erosion Subtraction) :

در این مرحله، مرزهای اشیاء از طریق تفریق تصویر erosion از ماسک باینری اصلی استخراج شده‌اند. erosion، لبه‌های بیرونی اشیاء را حذف می‌کند، و با تفریق آن از تصویر اصلی، تنها مرزهای نواحی سفید باقی می‌مانند. این روش برای جدا کردن مرزهای دقیق اشیاء و حذف نویز کاربرد دارد.

Perimeter of Objects (پیرامون اشیاء):

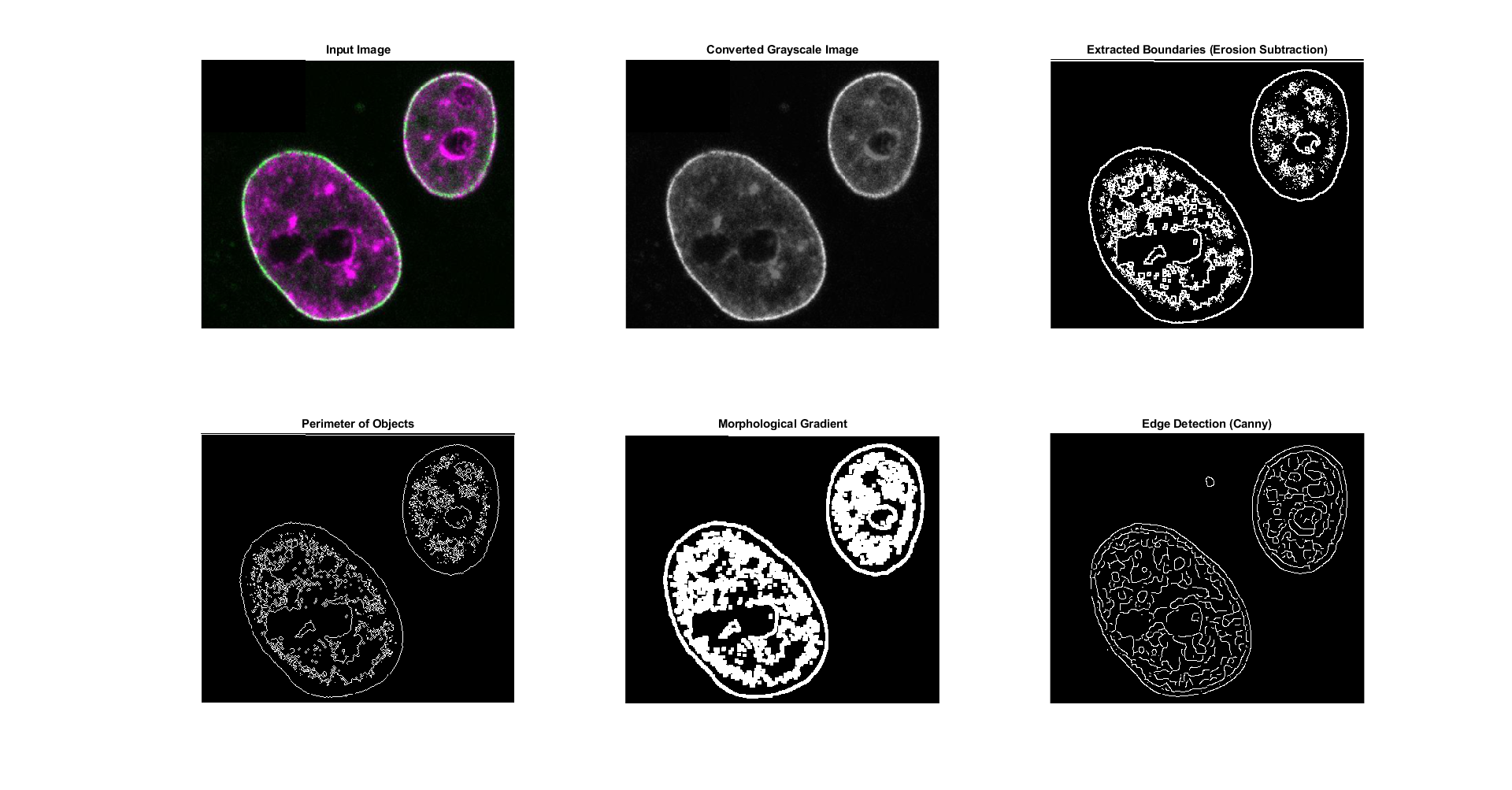
پیرامون اشیاء در تصویر باینری با استفاده از عملیات bwperim محاسبه شده است. این عملیات فقط پیکسل‌هایی را که در مرز نواحی سفید هستند نگه می‌دارد. این مرحله به استخراج دقیق‌تر خطوط خارجی اشیاء کمک می‌کند و برای اندازه‌گیری ویژگی‌هایی مانند محیط و شکل اجسام بسیار مفید است.

Morphological Gradient (گرادیان مورفولوژیکی):

گرادیان مورفولوژیکی با تفریق تصویر فرسایش‌یافته از تصویر Dilation به دست آمده است. این فرآیند اختلاف شدت بین لبه‌های بیرونی و داخلی اشیاء را برجسته می‌کند. گرادیان مورفولوژیکی برای شناسایی تغییرات شدت و مشخص کردن لبه‌های واضح در تصویر بسیار کاربردی است.

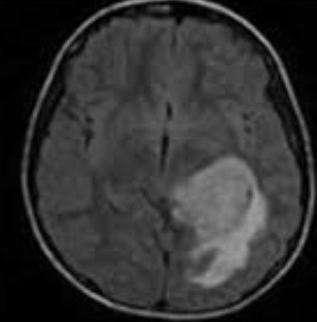
Edge Detection (Canny) (تشخیص لبه‌ها با Canny):

لبه‌های تصویر با استفاده از الگوریتم Canny شناسایی شده‌اند. این الگوریتم از فیلتر گوسین برای حذف نویز، محاسبه گرادیان تصویر برای تشخیص لبه‌ها، و اعمال آستانه‌های پایین و بالا برای جداسازی لبه‌های واقعی استفاده می‌کند. نتیجه این مرحله لبه‌های دقیق و پیوسته‌ای است که برای بخش‌بندی و تحلیل اشیاء در تصویر بسیار مهم هستند.



* 1. ناحیه‌بندی

1-3- با استفاده از روشهای segmentation ، شامل Otsu ، Thresholding و K-means Clustering، تومور موجود در تصویر زیر را ناحیه بندی کرده و نحوه ی عملکرد روشهای مختلف را مقایسه کنید.



1. Otsu Segmentation با 3 سطح (Filtered):

در این روش، ابتدا تصویر با استفاده از یک Gaussian Low-Pass Filter (GLPF) بهبود داده شده است تا نویز کاهش یابد و توزیع شدت روشنایی بهبود یابد. پس از اعمال فیلتر، الگوریتم Otsu thresholding برای بخش‌بندی تصویر به سه سطح شدت اعمال شد. نتیجه نشان‌دهنده جداسازی مناسب نواحی و برجسته‌سازی تومور با مرزهای واضح است. این روش به دلیل کاهش نویز، برای تحلیل نواحی با کنتراست پایین مؤثر است، اما ممکن است برخی جزئیات ساختاری ریز را از دست بدهد.

2. K-Means Segmentation با 4 Clusters:

K-Means clustering تصویر را بر اساس مقادیر شدت به چهار دسته مجزا تقسیم می‌کند. این روش پیکسل‌هایی با ویژگی‌های مشابه را گروه‌بندی کرده و تصویری با بخش‌بندی یکنواخت ایجاد می‌کند که نسبت به تغییرات جهانی شدت کمتر حساس است. با این حال، اگر درون تومور تغییرات شدت زیادی وجود داشته باشد، ممکن است باعث خطا در بخش‌بندی شود و برخی نواحی تومور به اشتباه به‌عنوان پس‌زمینه یا بافت‌های مجاور طبقه‌بندی شوند.

3. Otsu Segmentation با 3 سطح (Unfiltered):

در این روش، الگوریتم Otsu مستقیماً بر روی تصویر خاکستری اصلی اعمال شده است و تصویر را به سه سطح شدت تقسیم می‌کند. این روش مرزهای واضحی برای تومور فراهم می‌کند، اما نسبت به نویز حساس‌تر است. عدم استفاده از پیش‌پردازش منجر به ایجاد برخی نویزها و مصنوعات پراکنده شده است که قابلیت اطمینان این روش را در مجموعه داده‌های پر نویز کاهش می‌دهد.

4. Otsu Segmentation با 5 سطح:

با افزایش تعداد آستانه‌ها به پنج، الگوریتم Otsu نواحی با جزئیات بیشتری را تولید می‌کند و تغییرات شدت دقیق‌تر را ثبت می‌کند. اگرچه این روش می‌تواند اطلاعات بیشتری درباره بافت‌های اطراف ارائه دهد، ممکن است باعث بیش‌از‌حد بخش‌بندی تومور شود و تحلیل را پیچیده‌تر کند. این روش برای مواردی که نیاز به تمایز لایه‌های مختلف بافت یا اجزای متعدد است، مناسب می‌باشد.

خلاصه مقایسه:

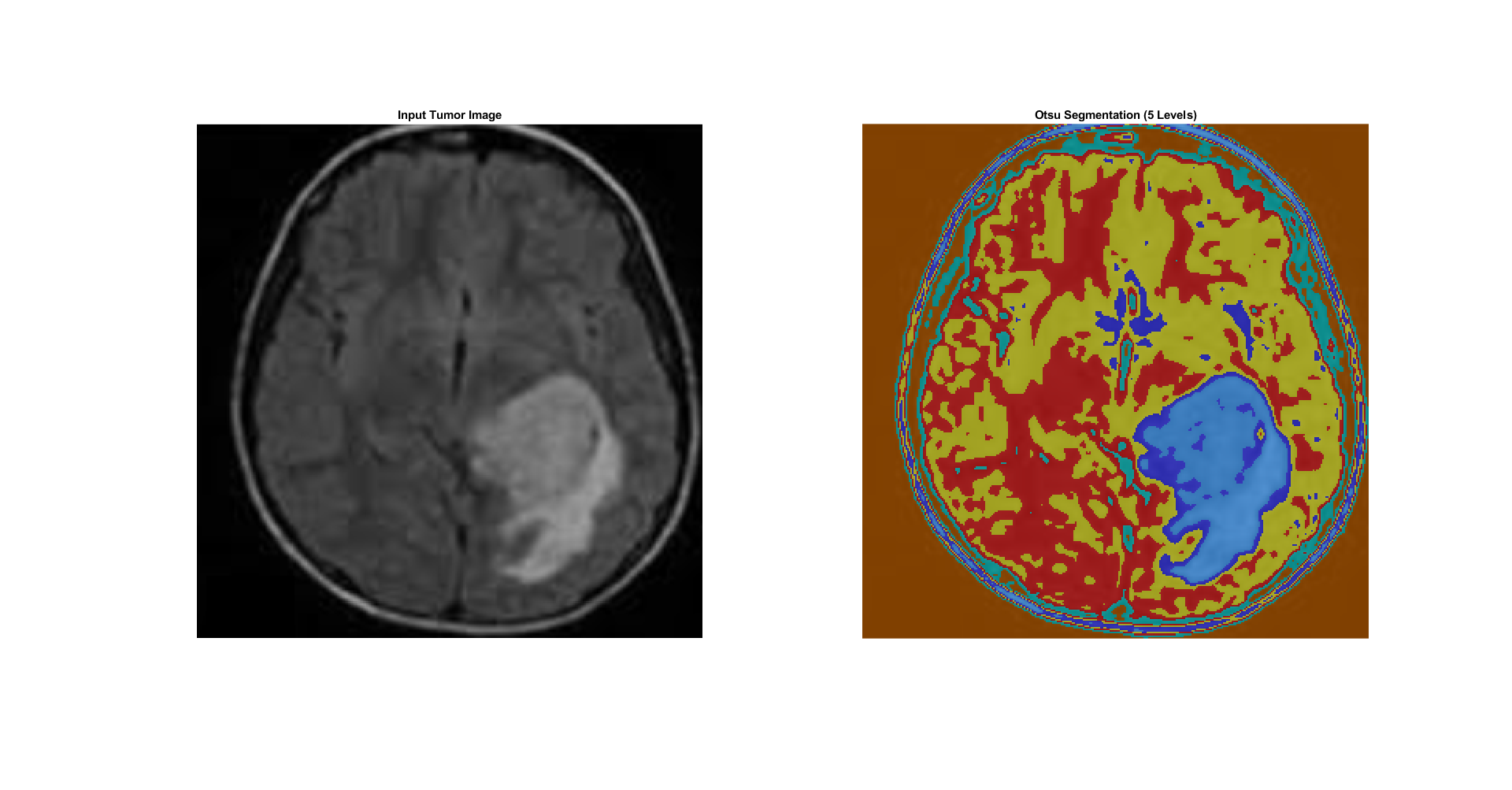
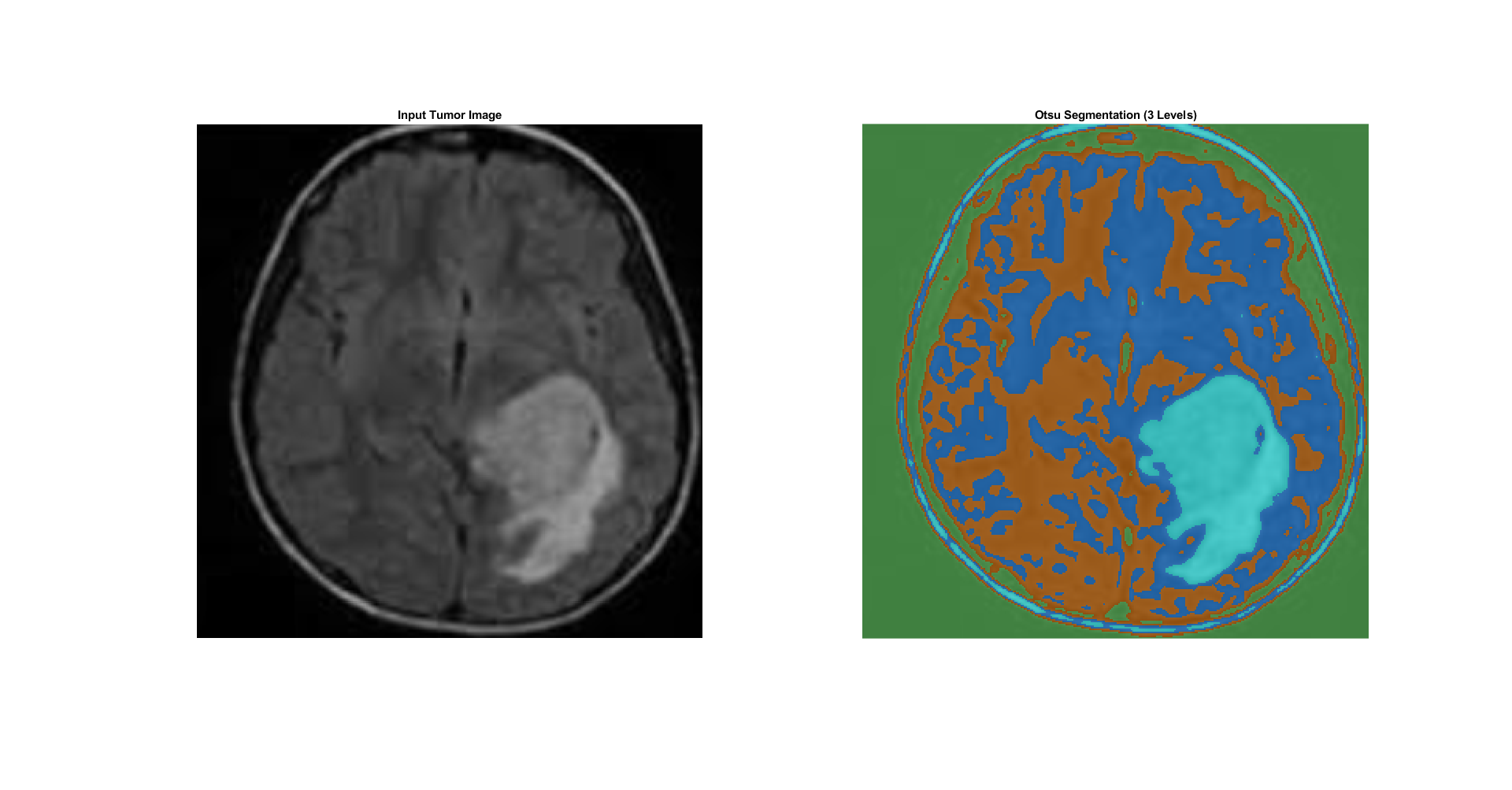
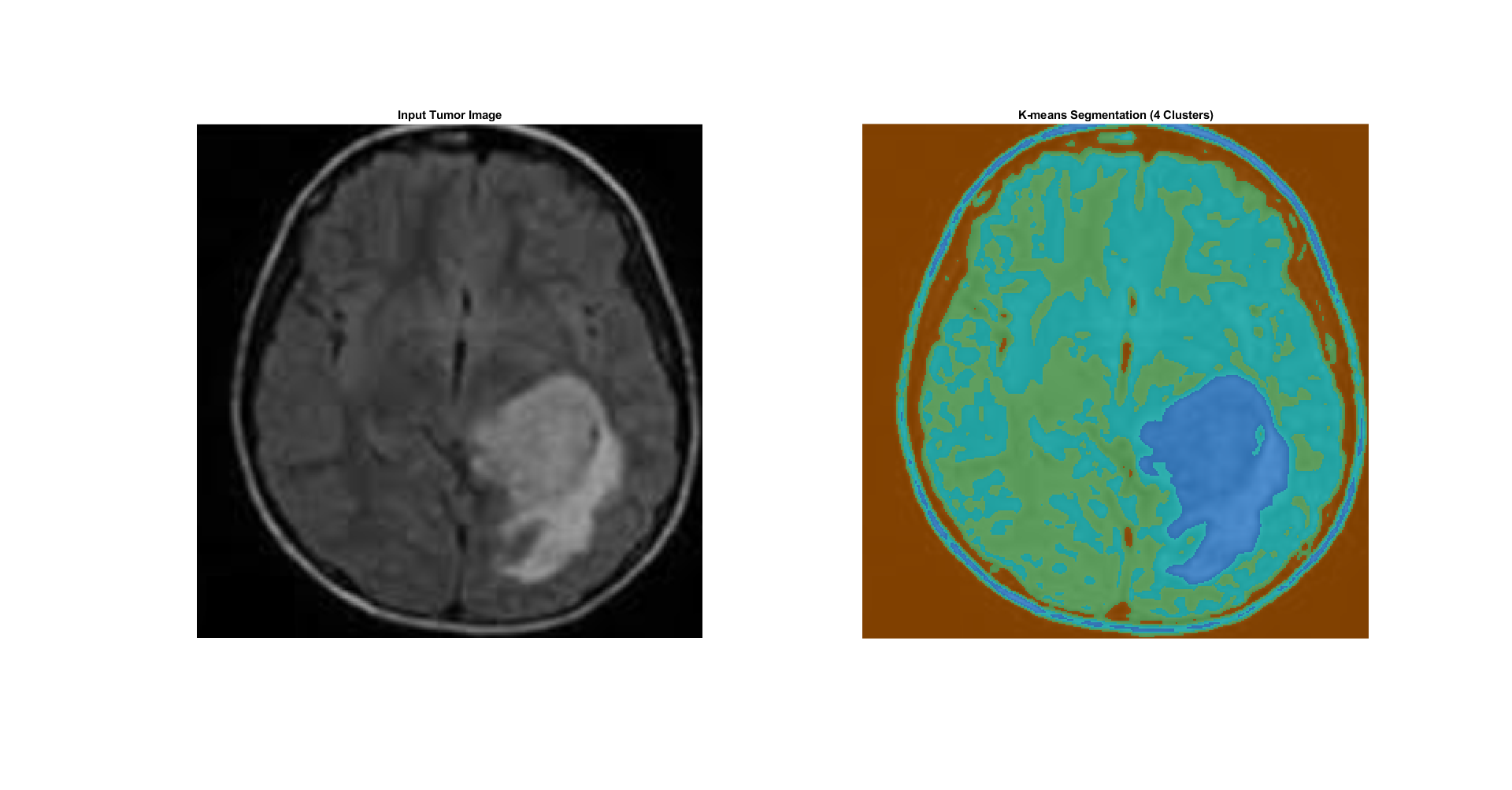
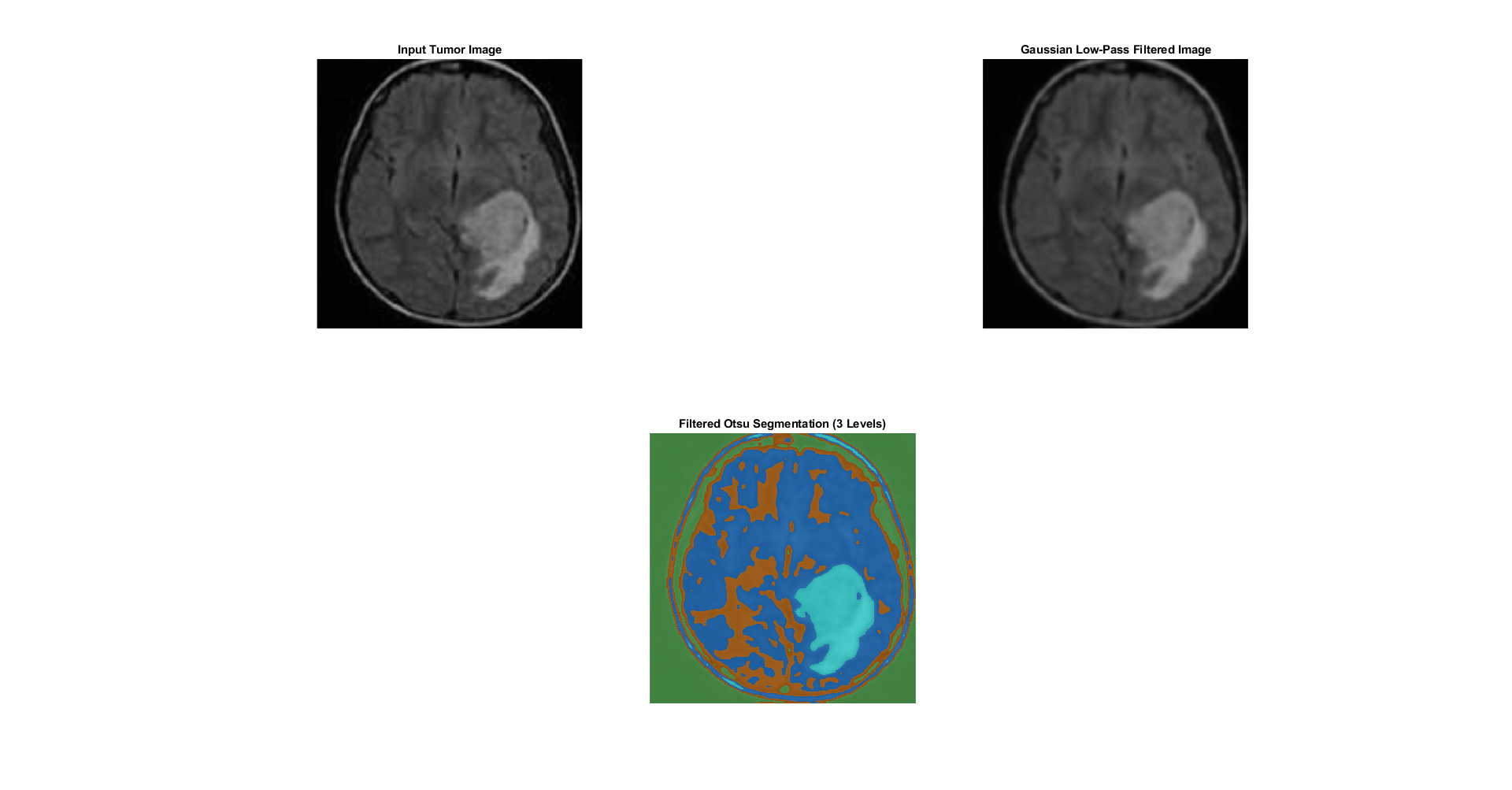
Filtered Otsu (3 Levels): بهترین گزینه برای کاهش نویز و شناسایی واضح ناحیه تومور با حواس‌پرتی کمتر از بافت‌های اطراف.

K-Means (4 Clusters): مناسب برای بخش‌بندی یکنواخت اما کمتر مقاوم در برابر تغییرات شدت درون تومور.

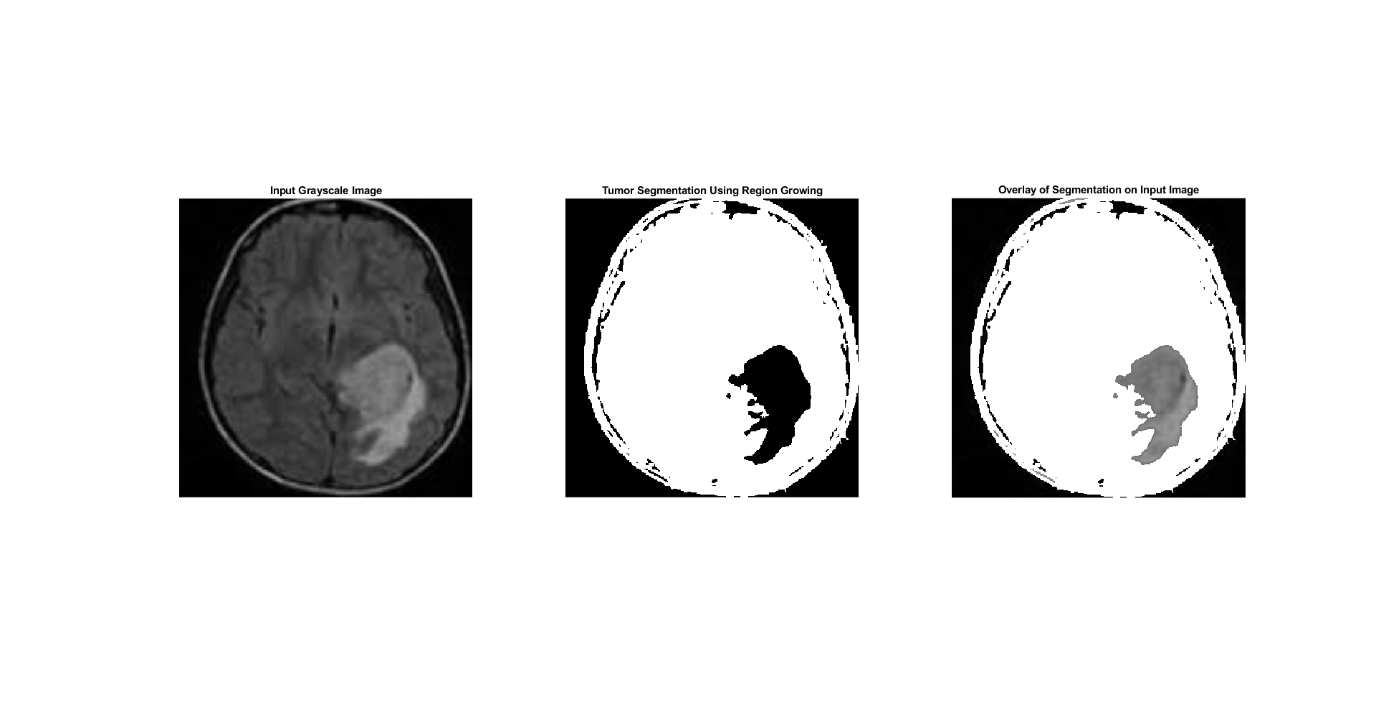
Otsu (3 Levels, Unfiltered): ساده و سریع اما مستعد نویز و مصنوعات، که قابلیت اطمینان آن را برای تحلیل‌های حساس کاهش می‌دهد.

Otsu (5 Levels): جزئیات بیشتری را ثبت می‌کند اما ممکن است باعث بخش‌بندی بیش‌از‌حد شده و جداسازی تومور را پیچیده کند.

برای بخش‌بندی تومور، روش Filtered Otsu با 3 سطح آستانه رویکرد متعادلی ارائه می‌دهد که نویز را کاهش داده و یک بخش‌بندی واضح و قابل تفسیر از ناحیه تومور حفظ می‌کند. در مواردی که نیاز به جزئیات بیشتر در بافت باشد، می‌توان از Otsu با 5 آستانه استفاده کرد، در حالی که K-Means برای خوشه‌بندی عمومی در شرایط با تغییرات محدود مناسب است.



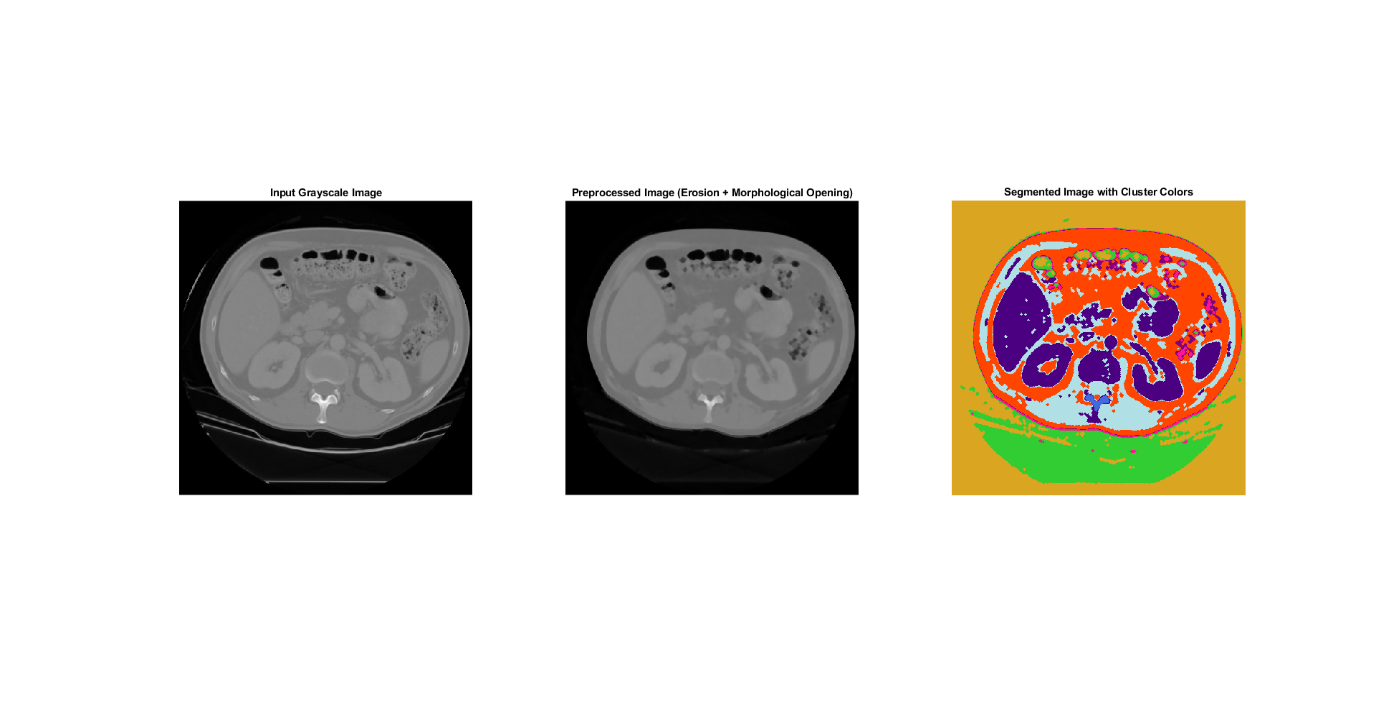
2-3- به کمک روش Region Growing، مساحت تومور را بدست آورید.



Tumor Area: 68906.5

* 1. تمرین 4

(۴) به کمک روشهای مورفولوژی اندامهای موجود در تصویر زیر را از هم جدا کرده و به کمک روش K-means Clustering، آنها را ناحیه بندی نمایید.



پس از لود کردن تصویر،morphological erosion با استفاده از یک disk-shaped structuring element اعمال می‌شود. این مرحله، جزئیات کوچک و نویزهای روشن را حذف کرده و یک پایه تمیزتر برای پردازش‌های بعدی ایجاد می‌کند. پس از آن، morphological opening برای صاف کردن نواحی، حذف ناهنجاری‌های کوچک و پر کردن شکاف‌های جزئی اعمال می‌شود. این مراحل پیش‌پردازش، تصویر را برای بخش‌بندی آماده کرده و انسجام ساختاری آن را افزایش می‌دهند.

در مرحله بعد، K-means clustering برای بخش‌بندی تصویر به 8 cluster مجزا بر اساس مقادیر شدت پیکسلی اعمال می‌شود. به هر cluster یک رنگ خاص اختصاص داده می‌شود که خروجی نهایی را به صورت یک تصویر بخش‌بندی‌شده و رنگی ارائه می‌دهد.

1. تمرین‌های مرتبط با کلاس حل تمرین
   1. تمرین 1

1. معماری شبکه‌ی U-Net را مطالعه کرده و خلاصه ای از ویژگیهای آن نوع مدل لایه ها و غیره) ارائه دهید.

U-Net به طور عمده برای وظایف تقسیم‌بندی تصویری معنایی (semantic image segmentation) طراحی شده است، جایی که هدف این است که هر پیکسل از تصویر به یکی از چندین دسته‌بندی اختصاص داده شود. این مدل برای اولین بار توسط اولاف رونن‌برگر و همکارانش در مقاله سال 2015 خود به نام U-Net: Convolutional Networks for Biomedical Image Segmentation معرفی شد. از آن زمان به یکی از مهم‌ترین مدل‌ها در زمینه تحلیل تصاویر پزشکی و سایر حوزه‌ها تبدیل شده است، به دلیل کارایی بالا در یادگیری از داده‌های محدود.

**ویژگی‌های کلیدی معماری U-Net:**

ساختار Encoder-Decoder: U-Net از معماری کلاسیک encoder-decoder پیروی می‌کند که به طور گسترده‌ای برای وظایف تقسیم‌بندی استفاده می‌شود. در این ساختار، مدل به طور تدریجی ابعاد فضایی تصویر ورودی را کاهش می‌دهد، در حالی که decoder تلاش می‌کند تا وضوح فضایی را دوباره به اندازه اولیه ورودی بازگرداند. این ساختار به مدل کمک می‌کند تا نمایه‌های غنی از ویژگی‌ها را به شکل نقشه‌های ویژگی یاد بگیرد، در حالی که زمینه‌های سطح بالا برای تقسیم‌بندی دقیق حفظ می‌شود.

1. **Encoder**: بخش encoder که به آن مسیر contracting نیز گفته می‌شود، شامل یک سری لایه‌های کانولوشنی است که وضوح فضایی تصویر ورودی را کاهش می‌دهند. این فرآیند به مدل کمک می‌کند تا ویژگی‌های انتزاعی و سطح بالا را در مقیاس‌های مختلف استخراج کند. به طور معمول در U-Net، encoder شامل بلوک‌های متوالی لایه‌های کانولوشنی به همراه لایه‌های max-pooling است.

* هر بلوک شامل دو لایه کانولوشن است که به دنبال آن تابع فعال‌سازی ReLU و normalization (batch normalization) اعمال می‌شود. لایه‌های کانولوشن معمولاً از کرنل 3x3 استفاده می‌کنند.
* پس از هر بلوک، نقشه ویژگی از طریق عملیات max-pooling با کرنل 2x2 نمونه‌برداری می‌شود.

1. **Decoder**: بخش decoder که به آن مسیر expansive نیز گفته می‌شود، هدف آن افزایش وضوح فضایی نقشه segmentation به اندازه تصویر ورودی است. این بخش از لایه‌های up-convolution یا transposed convolution استفاده می‌کند تا وضوح فضایی را افزایش دهد، با هدف دستیابی به پیش‌بینی‌های پیکسلی دقیق. در هر سطح، وضوح فضایی نقشه‌های ویژگی افزایش می‌یابد و ویژگی‌های یادگرفته شده در بخش encoder با دقت بیشتری تصحیح می‌شوند.

* هر بلوک در decoder شامل عملیات upsampling (از طریق transposed convolution) است که به دنبال آن با ویژگی‌های معادل از encoder ترکیب می‌شود، که این کار جزئیات با وضوح بالا را در طول فرآیند upscaling حفظ می‌کند.
* پس از عملیات upsampling، کانولوشن‌های اضافی برای پردازش بیشتر ویژگی‌ها اعمال می‌شود.

1. اتصالات **Skip** (ویژگی متمایز U-Net): یکی از متمایزترین ویژگی‌های معماری U-Net استفاده از اتصالات skip است. این اتصالات، خروجی بلوک‌های encoder را به طور مستقیم به بلوک‌های معادل در decoder متصل می‌کنند. این اتصالات به مدل این امکان را می‌دهند که ویژگی‌های فضایی کم‌سطح را از encoder با ویژگی‌های سطح بالاتر و انتزاعی‌تر از decoder ترکیب کند.

* این اتصالات به مدل اجازه می‌دهند که جزئیات دقیق تصویر ورودی را حفظ کند، که به ویژه برای وظایف segmentation که به تفکیک دقیق مرزهای اشیاء نیاز دارند، اهمیت دارد.
* ترکیب نقشه‌های ویژگی از encoder و decoder کمک می‌کند تا اطلاعات فضایی دقیق که معمولاً در عملیات downsampling در معماری‌های سنتی از دست می‌رود، حفظ شود.

1. **لایه‌های کانولوشن و فیلترها:** در معماری U-Net معمولاً از کانولوشن‌های 2D با اندازه کرنل 3x3 در هر دو بخش encoder و decoder استفاده می‌شود، که این کار مدل را کارآمد کرده و به آن این امکان را می‌دهد که ویژگی‌های محلی را یاد بگیرد. تعداد فیلترها در هر لایه به طور نمایی در مسیر encoder افزایش می‌یابد و در مسیر decoder به طور معکوس کاهش می‌یابد.

* مسیر encoder ممکن است با 64 فیلتر در اولین بلوک کانولوشن آغاز شود، که به تدریج به 128، 256 و غیره افزایش می‌یابد، در حالی که مسیر decoder از بزرگ‌ترین تعداد فیلترها شروع شده و به طور متقارن کاهش می‌یابد.

1. **لایه خروجی:** لایه خروجی U-Net معمولاً شامل یک کانولوشن 1x1 است که عمق نقشه ویژگی را به تعداد کلاس‌های مورد نظر (برای segmentation چندکلاسه) یا یک کانال (برای segmentation دوتایی) کاهش می‌دهد. نتیجه این لایه یک نقشه segmentation است که در آن هر پیکسل به یک برچسب کلاس اختصاص داده می‌شود.

* در وظایف segmentation دوتایی معمولاً از تابع فعال‌سازی sigmoid در لایه خروجی استفاده می‌شود، در حالی که برای segmentation چندکلاسه از تابع فعال‌سازی softmax استفاده می‌شود، که تضمین می‌کند پیش‌بینی‌های پیکسلی برای هر کلاس جمعاً برابر با 1 باشد.

1. **تابع هزینه (Loss Function):** برای وظایف segmentation، U-Net معمولاً از یک تابع هزینه پیکسلی استفاده می‌کند. انتخاب‌های رایج عبارتند از:

* تابع هزینه cross-entropy (برای segmentation چندکلاسه).
* ضریب Dice یا Dice similarity coefficient (DSC)، که به طور ویژه برای داده‌های نامتعادل یا تصاویر پزشکی مناسب است، زیرا به طور مستقیم هم‌پوشانی بین segmentation پیش‌بینی شده و segmentation حقیقت زمینه را اندازه‌گیری می‌کند.
* cross-entropy وزنی ممکن است در مواقعی که کلاس‌ها بسیار نامتعادل هستند، اعمال شود.

1. **داده افزایی (Data Augmentation):** مدل‌های U-Net، به ویژه در زمینه segmentation تصاویر پزشکی، معمولاً روی داده‌های نسبتاً کوچکی آموزش داده می‌شوند. برای کاهش overfitting و افزایش کارایی مدل، تکنیک‌های تقویت داده زیادی به کار می‌روند. این‌ها می‌توانند شامل موارد زیر باشند:

* چرخش، انتفال، مقیاس‌دهی برای ایجاد تنوع در داده‌های آموزشی.
* فلیپ کردن و تغییرات الاستیک برای شبیه‌سازی اعوجاجات دنیای واقعی.
* برش تصادفی و تغییر مقیاس برای تمرکز بر روی نواحی مختلف تصویر.

**مزایای U-Net:**

**Localization دقیق:** اتصالات skip و معماری encoder-decoder به U-Net این امکان را می‌دهند که ویژگی‌های دقیق را حفظ کرده و ویژگی‌های سطح بالا را در حالی که جزئیات را یاد می‌گیرد، در کنار هم داشته باشد، که این ویژگی را برای وظایف نیازمند تقسیم‌بندی دقیق پیکسلی بسیار مؤثر می‌کند.

**آموزش کارآمد:** معماری متقارن U-Net با اتصالات skip به آن اجازه می‌دهد که با داده‌های آموزشی کمتر نیز عملکرد خوبی داشته باشد، که این ویژگی به ویژه در زمینه‌های مانند تحلیل تصاویر پزشکی که داده‌های برچسب‌گذاری شده محدود هستند، مفید است.

**انعطاف‌پذیری:** U-Net به راحتی می‌تواند به وظایف مختلف segmentation غیر از تصاویر پزشکی، مانند تقسیم‌بندی تصاویر ماهواره‌ای، تحلیل‌های کشاورزی و حتی برخی از وظایف تصاویر طبیعی اعمال شود.

چالش‌ها و نسخه‌های توسعه یافته: در حالی که U-Net یک معماری قدرتمند برای segmentation است، برخی چالش‌ها نیز وجود دارد:

**Overfitting**: به دلیل تعداد بالای پارامترها، U-Net مستعد overfitting است، به ویژه در داده‌های کوچک. استفاده از تکنیک‌های منظم‌سازی مانند dropout، تقویت داده‌ها و توقف زودهنگام ضروری است.

**هزینه محاسباتی:** U-Net می‌تواند از نظر محاسباتی پرهزینه باشد، به ویژه برای تصاویر با وضوح بالا، زیرا decoder باید نقشه‌های ویژگی با ابعاد بزرگتر را پردازش کند.

نسخه‌های مختلفی از U-Net برای رفع این محدودیت‌ها پیشنهاد شده است:

3D U-Net: برای داده‌های حجمی (مثل اسکن‌های CT و MRI)، از کانولوشن‌های 3D به جای 2D استفاده می‌شود تا بعد اضافی را مدیریت کند.

Attention U-Net: مکانیزم‌های توجه را به معماری اضافه می‌کند تا به شبکه اجازه دهد بر بخش‌های مرتبط‌تر از تصویر تمرکز کند و عملکرد آن را در وظایف پیچیده‌تر تقسیم‌بندی بهبود بخشد.

(۲) حداقل چهار مورد از هایپر پارامترهای مدل پیوست شده (تعداد epoch ها تعداد لایه ها، نرخ تعلیم، سایز بچ و غیره) را برای سه مقدار متفاوت، تغییر دهید و اثر هر کدام از آنها را روی عملکرد مدل دقت، خروجی و غیره)، بررسی نمایید. (در صورت تمایل میتوانید از کد مثال U-Net بررسی شده در کلاس یا کدهای دیگری که از مدل U-Net استفاده کرده‌اند نیز استفاده نمایید.)

پارامتر‌های Epoch, BatchSize, LearningRate و تعدادلایه ها (Filter) برای جمعاً 81 حالت مختلف بررسی شده و در نهایت همه‌ اطلاعات در یک csv ذخیره شده است که Best\_Val\_Accuracy نشان دهنده بهترین دقت هر کدام هست.

**بهترین دقت:**

بالاترین دقت اعتبار سنجی با پیکربندی به دست آمد:

* دوره ها: 10
* حجم دسته: 8
* میزان یادگیری: 0.001
* فیلترها: [32، 64، 128، 256، 512]
* بهترین دقت Val: 0.9653 (مسیر مدل:

models/UNet\_e10\_bs8\_lr0.001\_f32\_64\_128\_256\_512.keras)

این پیکربندی بالاترین دقت را در میان سایر hyperparameters ارائه می‌دهد، که نشان می‌دهد استفاده از اندازه دسته‌ای 8، نرخ یادگیری 0.001، و مجموعه بزرگ‌تری از فیلترها (یعنی [32، 64، 128، 256، 512]) منجر به بهترین عملکرد مدل.

**تاثیر نرخ یادگیری:**

* نرخ یادگیری 0.001 به طور مداوم نتایج بهتری را در مقایسه با 0.0001 و 0.01 در اکثر پیکربندی ها به همراه داشت. به عنوان مثال:
* نرخ یادگیری 0.001 با فیلترهای [32، 64، 128، 256، 512] و اندازه دسته 8 منجر به بالاترین دقت اعتبارسنجی 0.9653 شد.
* نرخ یادگیری 0.0001 با همان مجموعه فیلتر، دقت اعتبار 0.9656 را ارائه می دهد، بسیار نزدیک اما کمی بهتر از نرخ 0.001.
* نرخ یادگیری 0.01 منجر به عملکرد پایین‌تر در پیکربندی‌های مختلف فیلتر، با دقت اعتبارسنجی در حدود 0.91 و کمتر شد.

**دوره یا Epoch:**

* تقریباً در هر مورد، 10 دوره نتایج بهتری نسبت به 5 دوره ارائه کردند، که نشان می‌دهد آموزش طولانی‌تر به مدل اجازه می‌دهد تا داده‌ها را بهتر بهینه کند و از آن یاد بگیرد. به عنوان مثال، با 10 دوره و اندازه دسته 8، بهترین دقت به 0.9653 رسید.
* حالت‌هایی با 5 دوره آموزش، نتایج خوبی نشان دادند، اما دقت تمایل به تثبیت یا کاهش با مجموعه فیلترهای کوچکتر و نرخ یادگیری بالاتر (0.01) داشت.

**اندازه دسته:**

* اندازه دسته 8 در بسیاری از موارد بهتر از 2 و 16 عمل می کند، به خصوص زمانی که با نرخ یادگیری 0.001 و فیلترها جفت شوند [32، 64، 128، 256، 512]. به عنوان مثال، با 10 دوره، اندازه دسته 8 منجر به بالاترین دقت اعتبارسنجی (0.9653) شد.
* اندازه دسته 16 نتایج متفاوتی داشت، با حداکثر دقت 0.9479 برای فیلترها [32، 64، 128، 256، 512]، و مابقیه پیکربندی‌ها دقت کمتری را نشان دادند. به عنوان مثال، اندازه دسته 16 با نرخ یادگیری 0.01 و فیلترهای [32، 64، 128، 256، 512] کاهش شدیدی را به 0.8355 نشان دادند.

**مجموعه فیلتر:**

* مجموعه فیلترهای بزرگتر مانند [32، 64، 128، 256، 512] به طور مداوم در همه پیکربندی ها عملکرد بهتری داشتند. این از بالاترین دقت اعتبار سنجی به دست آمده با این مجموعه فیلترها، مانند 0.9653 برای 10 دوره، اندازه دسته 8 و نرخ یادگیری 0.001 مشهود است.
* مجموعه فیلترهای کوچکتر، مانند [8، 16، 32، 64، 128]، منجر به عملکرد پایین تر، به ویژه با نرخ یادگیری 0.01 شد.

**نتیجه گیری:**

بهترین پیکربندی: بهترین تنظیمات این است:

* دوره ها: 10
* حجم دسته: 8
* میزان یادگیری: 0.001
* فیلترها: [32، 64، 128، 256، 512] این پیکربندی به بالاترین دقت اعتبارسنجی 0.9653 دست یافت.

**پارامترهایی که بیشترین تاثیر را دارند:**

* نرخ یادگیری: به نظر می رسد مقدار 0.001 بهینه ترین باشد.
* اندازه فیلتر: مجموعه فیلترهای بزرگتر مانند [32، 64، 128، 256، 512] منجر به استخراج بهتر ویژگی ها و عملکرد بالاتر مدل می شود.
* دوره ها و اندازه دسته ای: افزایش دوره ها به 10 و استفاده از اندازه دسته ای 8 به همگرایی بیشتر مدل کمک کرد.

اثر هر کدام از پارامتر‌های دستکاری شده روی مدل به صورت نمودار آورده شده است.

A diagram of different colored rectangular shapes

AI-generated content may be incorrect.

A diagram of different colored squares

AI-generated content may be incorrect.

A graph showing a comparison of a number of objects

AI-generated content may be incorrect.

A graph showing different colored squares

AI-generated content may be incorrect.

در نهایت خروجی بهترین مدل برای داده تست به شکل زیر است:

A collage of images of white spots

AI-generated content may be incorrect.

جدول زیر برای بررسی دقیق تر تمامی حالت ها است:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epochs | Batch\_Size | Learning\_Rate | Filters | Best\_Val\_Accuracy |
| 5 | 2 | 0.0001 | [8, 16, 32, 64, 128] | 0.836277723 |
| 5 | 2 | 0.0001 | [16, 32, 64, 128, 256] | 0.92788583 |
| 5 | 2 | 0.0001 | [32, 64, 128, 256, 512] | 0.93643868 |
| 5 | 2 | 0.001 | [8, 16, 32, 64, 128] | 0.959655762 |
| 5 | 2 | 0.001 | [16, 32, 64, 128, 256] | 0.943916619 |
| 5 | 2 | 0.001 | [32, 64, 128, 256, 512] | 0.942704916 |
| 5 | 2 | 0.01 | [8, 16, 32, 64, 128] | 0.895415127 |
| 5 | 2 | 0.01 | [16, 32, 64, 128, 256] | 0.919727445 |
| 5 | 2 | 0.01 | [32, 64, 128, 256, 512] | 0.835535109 |
| 5 | 8 | 0.0001 | [8, 16, 32, 64, 128] | 0.836095154 |
| 5 | 8 | 0.0001 | [16, 32, 64, 128, 256] | 0.836095154 |
| 5 | 8 | 0.0001 | [32, 64, 128, 256, 512] | 0.847661316 |
| 5 | 8 | 0.001 | [8, 16, 32, 64, 128] | 0.836095154 |
| 5 | 8 | 0.001 | [16, 32, 64, 128, 256] | 0.941042602 |
| 5 | 8 | 0.001 | [32, 64, 128, 256, 512] | 0.938610077 |
| 5 | 8 | 0.01 | [8, 16, 32, 64, 128] | 0.916594207 |
| 5 | 8 | 0.01 | [16, 32, 64, 128, 256] | 0.836095154 |
| 5 | 8 | 0.01 | [32, 64, 128, 256, 512] | 0.86239624 |
| 5 | 16 | 0.0001 | [8, 16, 32, 64, 128] | 0.836087525 |
| 5 | 16 | 0.0001 | [16, 32, 64, 128, 256] | 0.836095154 |
| 5 | 16 | 0.0001 | [32, 64, 128, 256, 512] | 0.836095154 |
| 5 | 16 | 0.001 | [8, 16, 32, 64, 128] | 0.862201691 |
| 5 | 16 | 0.001 | [16, 32, 64, 128, 256] | 0.918136597 |
| 5 | 16 | 0.001 | [32, 64, 128, 256, 512] | 0.876190186 |
| 5 | 16 | 0.01 | [8, 16, 32, 64, 128] | 0.836095154 |
| 5 | 16 | 0.01 | [16, 32, 64, 128, 256] | 0.836095154 |
| 5 | 16 | 0.01 | [32, 64, 128, 256, 512] | 0.836095154 |
| 10 | 2 | 0.0001 | [8, 16, 32, 64, 128] | 0.917741537 |
| 10 | 2 | 0.0001 | [16, 32, 64, 128, 256] | 0.961669922 |
| 10 | 2 | 0.0001 | [32, 64, 128, 256, 512] | 0.959838867 |
| 10 | 2 | 0.001 | [8, 16, 32, 64, 128] | 0.960998535 |
| 10 | 2 | 0.001 | [16, 32, 64, 128, 256] | 0.958007813 |
| 10 | 2 | 0.001 | [32, 64, 128, 256, 512] | 0.965332031 |
| 10 | 2 | 0.01 | [8, 16, 32, 64, 128] | 0.888345242 |
| 10 | 2 | 0.01 | [16, 32, 64, 128, 256] | 0.918202698 |
| 10 | 2 | 0.01 | [32, 64, 128, 256, 512] | 0.835535109 |
| 10 | 8 | 0.0001 | [8, 16, 32, 64, 128] | 0.836095154 |
| 10 | 8 | 0.0001 | [16, 32, 64, 128, 256] | 0.836095154 |
| 10 | 8 | 0.0001 | [32, 64, 128, 256, 512] | 0.943647087 |
| 10 | 8 | 0.001 | [8, 16, 32, 64, 128] | 0.949820399 |
| 10 | 8 | 0.001 | [16, 32, 64, 128, 256] | 0.950892866 |
| 10 | 8 | 0.001 | [32, 64, 128, 256, 512] | 0.950509191 |
| 10 | 8 | 0.01 | [8, 16, 32, 64, 128] | 0.932458222 |
| 10 | 8 | 0.01 | [16, 32, 64, 128, 256] | 0.836095154 |
| 10 | 8 | 0.01 | [32, 64, 128, 256, 512] | 0.836095154 |
| 10 | 16 | 0.0001 | [8, 16, 32, 64, 128] | 0.836092651 |
| 10 | 16 | 0.0001 | [16, 32, 64, 128, 256] | 0.836095154 |
| 10 | 16 | 0.0001 | [32, 64, 128, 256, 512] | 0.836095154 |
| 10 | 16 | 0.001 | [8, 16, 32, 64, 128] | 0.836095154 |
| 10 | 16 | 0.001 | [16, 32, 64, 128, 256] | 0.94237411 |
| 10 | 16 | 0.001 | [32, 64, 128, 256, 512] | 0.947865784 |
| 10 | 16 | 0.01 | [8, 16, 32, 64, 128] | 0.836095154 |
| 10 | 16 | 0.01 | [16, 32, 64, 128, 256] | 0.836095154 |
| 10 | 16 | 0.01 | [32, 64, 128, 256, 512] | 0.836095154 |
| 20 | 2 | 0.0001 | [8, 16, 32, 64, 128] | 0.956237793 |
| 20 | 2 | 0.0001 | [16, 32, 64, 128, 256] | 0.963867188 |
| 20 | 2 | 0.0001 | [32, 64, 128, 256, 512] | 0.965637207 |
| 20 | 2 | 0.001 | [8, 16, 32, 64, 128] | 0.95703125 |
| 20 | 2 | 0.001 | [16, 32, 64, 128, 256] | 0.957336426 |
| 20 | 2 | 0.001 | [32, 64, 128, 256, 512] | 0.957580566 |
| 20 | 2 | 0.01 | [8, 16, 32, 64, 128] | 0.907021999 |
| 20 | 2 | 0.01 | [16, 32, 64, 128, 256] | 0.908963799 |
| 20 | 2 | 0.01 | [32, 64, 128, 256, 512] | 0.835535109 |
| 20 | 8 | 0.0001 | [8, 16, 32, 64, 128] | 0.836095154 |
| 20 | 8 | 0.0001 | [16, 32, 64, 128, 256] | 0.836095154 |
| 20 | 8 | 0.0001 | [32, 64, 128, 256, 512] | 0.952349007 |
| 20 | 8 | 0.001 | [8, 16, 32, 64, 128] | 0.953892291 |
| 20 | 8 | 0.001 | [16, 32, 64, 128, 256] | 0.95356971 |
| 20 | 8 | 0.001 | [32, 64, 128, 256, 512] | 0.951843262 |
| 20 | 8 | 0.01 | [8, 16, 32, 64, 128] | 0.836134613 |
| 20 | 8 | 0.01 | [16, 32, 64, 128, 256] | 0.940110505 |
| 20 | 8 | 0.01 | [32, 64, 128, 256, 512] | 0.836095154 |
| 20 | 16 | 0.0001 | [8, 16, 32, 64, 128] | 0.836095154 |
| 20 | 16 | 0.0001 | [16, 32, 64, 128, 256] | 0.836095154 |
| 20 | 16 | 0.0001 | [32, 64, 128, 256, 512] | 0.836095154 |
| 20 | 16 | 0.001 | [8, 16, 32, 64, 128] | 0.951520622 |
| 20 | 16 | 0.001 | [16, 32, 64, 128, 256] | 0.954956055 |
| 20 | 16 | 0.001 | [32, 64, 128, 256, 512] | 0.95272392 |
| 20 | 16 | 0.01 | [8, 16, 32, 64, 128] | 0.893404663 |
| 20 | 16 | 0.01 | [16, 32, 64, 128, 256] | 0.913698852 |
| 20 | 16 | 0.01 | [32, 64, 128, 256, 512] | 0.836095154 |