

بسمه تعالی



آزمایشگاه پردازش سیگنال و تصاویر پزشکی

آزمایش نهم؛ ناحیه بندی تصاویر پزشکی

نام استاد:

دکتر سپیده حاجی پور

نام دانشجویان:

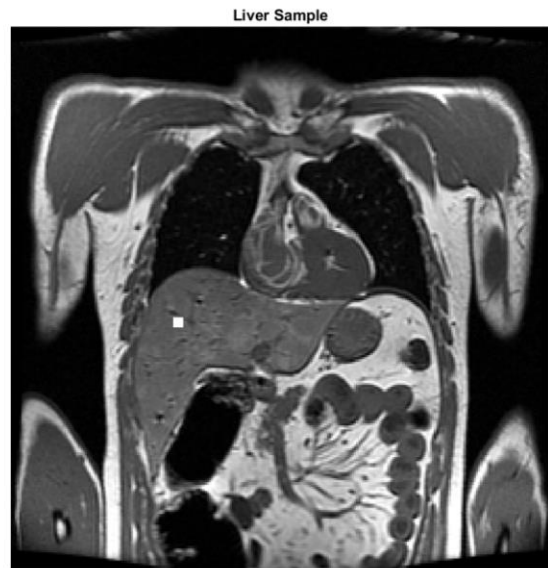
مهدی نوروزی ۹۷۱۰۲۵۹۳

آرمین نوردی ۹۹۱۰۵۱۲۹

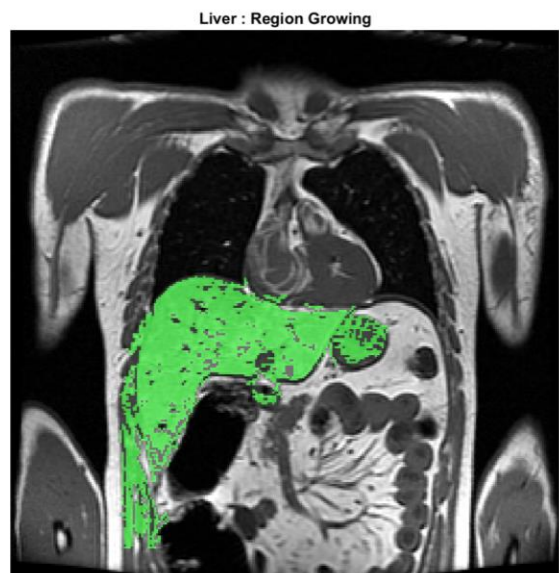
تاریخ تحویل:

۱۴۰۲/۱۰/۱۳

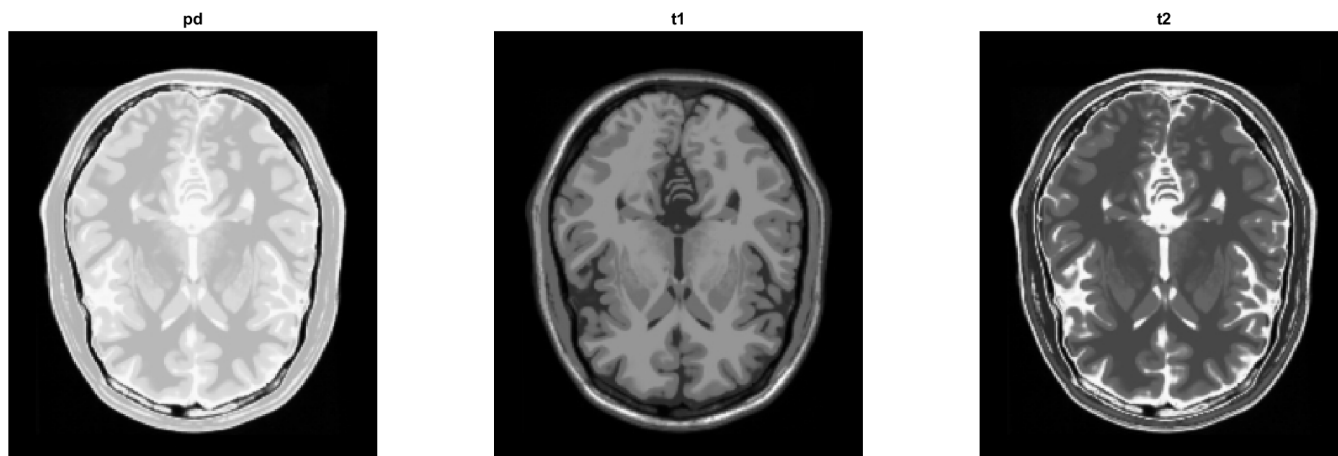
(Q1) با آزمون و خطا مختصات سه پیکسل نمونه برای ریه‌ها و کبد انتخاب کرده و اطراف آن‌ها یک مربع با ابعاد ۵ پیکسل به عنوان مجموعه‌ی مرجع در نظر گرفته و نمایش می‌دهیم (میانگین روشنایی خانه‌های هر مربع به عنوان معیار مقایسه‌ی آن اندام تعیین می‌شود):



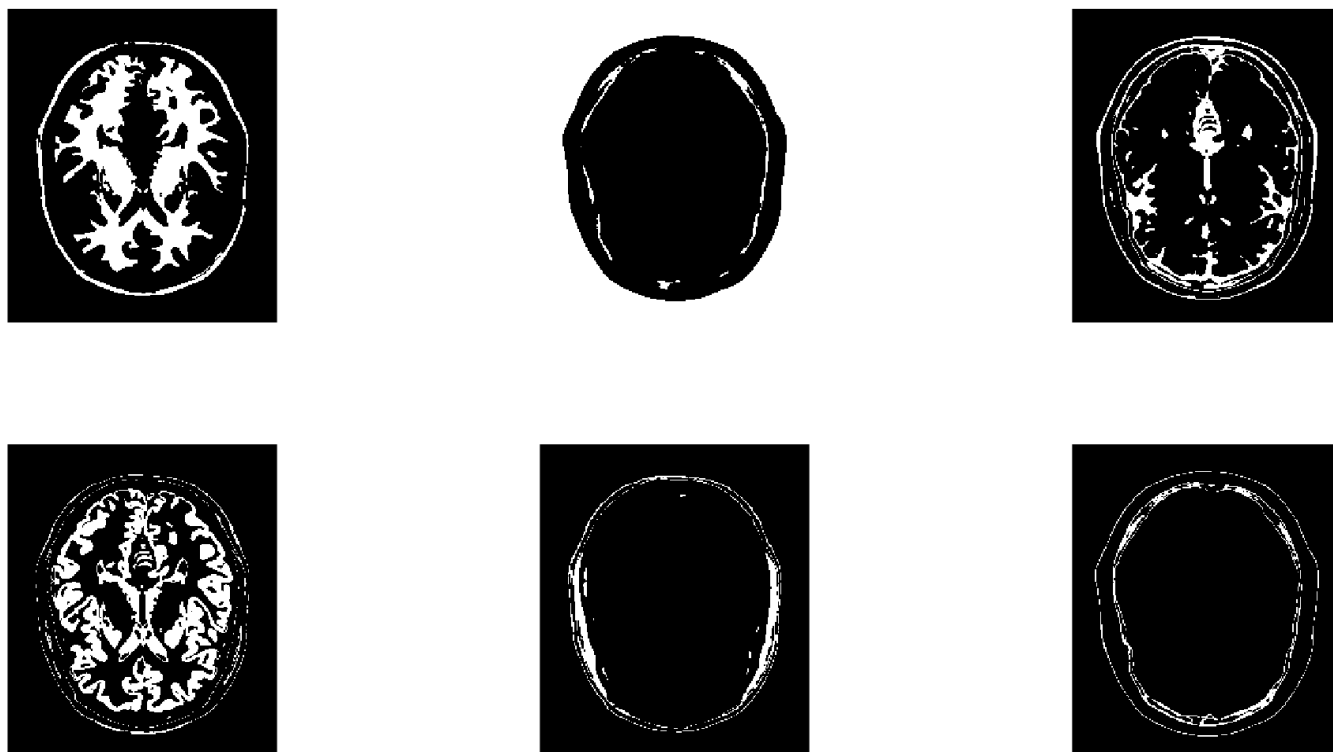
برای پیاده‌سازی روش Region Growing کافیست در ماسک ابتدایی همه‌ی خانه‌ها به جز خانه‌های نمونه را صفر قرار دهیم. سپس در هر مرحله، همسایه‌های تمامی خانه‌های تاییدشده‌ی جدید را بررسی کرده و در صورت قرار گرفتن روشنایی‌شان در بازه‌ی مدنظر، آن‌ها را نیز تایید و در غیراین‌صورت، رد کنیم. این کار تا زمانی که دیگر خانه‌ی تاییدشده‌ی جدیدی شناسایی نشود ادامه می‌یابد. می‌بینیم که این روش برای ریه‌ها بسیار خوب عمل کرده، چون روشنایی ریه‌ها از بافت‌های مجاورش متمایز است و حتی با گرفتن یک بازه‌ی آستانه‌ی بسیار بزرگ هم هنگام بررسی همسایه‌ها دچار خطا نمی‌شویم. این در حالیست که برای کبد که دارای روشنایی مشابه با بخش‌هایی از قلب و معده است، اندکی بزرگ گرفتن بازه باعث اشتباه سیستم و حرکت به بافت‌های مجاور می‌شود. به علاوه این آستانه‌ی سخت‌گیرانه به از دست دادن بخش‌های تیره‌تر درون کبد نیز می‌شود.



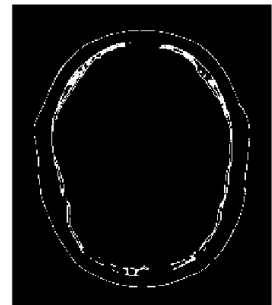
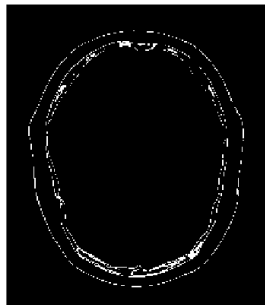
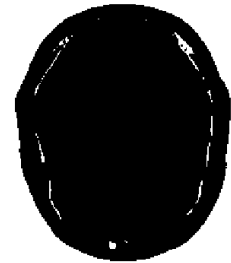
Q2 ابتدا سه تصویر را کنار هم قرار داده و پیکسل‌های تصویر را از حالت دوبعدی به صورت برداری درمی‌آوریم به شکلی که در نهایت به تعداد نقاط تصویر اولیه، داده‌های سه‌مولفه‌ای (هریک برابر روشنایی یکی از سه تصویر) داشته باشیم.



سپس الگوریتم kmeans متلب را با ۶ خوشه اجرا کرده و به هریک از نقاط تصویر، شماره‌ی یکی از خوشه‌ها را نسبت می‌دهیم. در نهایت با جداسازی داده‌ها و نشان دادن آن‌ها با رنگ سفید، به خوشه‌های زیر می‌رسیم. با مقایسه‌ی این تصاویر با تصاویر اولیه‌ی بالا می‌توان مشاهده کرد که تقسیم‌بندی به خوبی انجام شده و تصویر را به دسته‌های قشر سفید، قشر خاکستری، فضای subarachnoid، مرزهای مجسمه و فضای بیرونی سر تفکیک کرده‌است. البته باید توجه کرد که با هر بار اجرای این برنامه، به دلیل تخصیص تصادفی مرکزهای اولیه‌ی خوشه‌ها، ممکن است دسته‌بندی‌های اندکی متفاوت ببینیم. پس لازم است با تکرار، خوشه‌بندی مطلوب را بیابیم.



Q3) برای پیاده‌سازی دستی الگوریتم kmeans پس از آماده‌سازی داده‌ها باید برای هر یک از خوشه‌ها به تصادف یکی از داده‌ها را انتخاب کرده و به عنوان مرکز اولیه‌ی آن خوشه قرار دهیم. سپس در هر مرحله، نزدیک‌ترین داده‌ها به هر مرکز را به آن خوشه نسبت داده و سپس مرکزها را برحسب داده‌های جدید خوشه‌ها به‌روز می‌کنیم. نتیجه‌ی نهایی ۶ خوشه در الگوریتم دستی به صورت زیر است. می‌بینیم که این تصاویر بسیار مشابه الگوریتم متلب تولید شده‌اند و خوشه‌بندی را به‌خوبی انجام داده‌اند (در اینجا نیز مسئله‌ی وابستگی به تکرار و تخصیص تصادفی مراکز اولیه برقرار است).



Q4) الگوریتم fcm را نیز مشابه kmeans با دستور متلب اجرا می‌کنیم. البته در انتها با پیمایش داده‌ها، شماره‌ی خوشه‌ای که بیشترین مقدار u (احتمال) را دارد به هر داده نسبت می‌دهیم. نتایج در این روش اندکی بهتراند زیرا پیوستگی بین داده‌های هر خوشه بیشتر است و داده‌های هر خوشه از یک جنس هستند (برای مثال، به نظر می‌رسد که این روش توانسته مرزهای dura را هم تا حد خوبی جدا کرده و نمایش دهد).

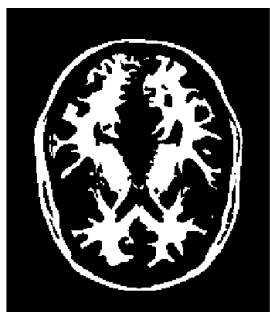
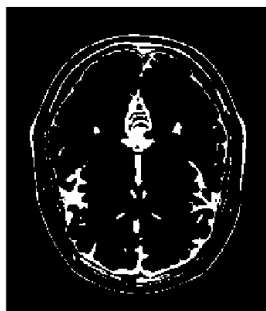
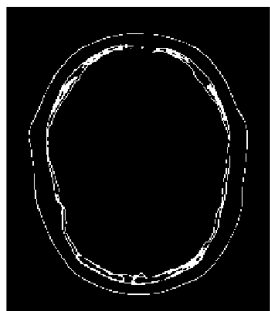
اثبات: ابتدا با افزودن شرط مجموع ضرایب (برای هر داده) معادله‌ی لاگرانژ را تشکیل داده و سپس نسبت به هریک از سه متغیر موجود مشتق گرفته و برابر صفر قرار می‌دهیم:

$$L = \sum_{i=1}^D \sum_{j=1}^N \mu_{ij}^m \|x_i - c_j\|^2 - \sum_{i=1}^D \lambda_i \left(\sum_{j=1}^N \mu_{ij} - 1 \right)$$

$$\frac{\partial L}{\partial c_j} = \sum_{i=1}^D 2\mu_{ij}^m (c_j - x_i) = 0 \quad \Rightarrow \quad c_j = \frac{\sum_{i=1}^D \mu_{ij}^m x_i}{\sum_{i=1}^D \mu_{ij}^m}, \quad \frac{\partial L}{\partial \lambda_j} = \sum_{j=1}^N \mu_{ij} - 1 = 0$$

$$\frac{\partial L}{\partial \mu_{ij}} = m\mu_{ij}^{m-1} \|x_i - c_j\|^2 - \lambda_i = 0 \quad \Rightarrow \quad \mu_{ij} = \left(\frac{\lambda_i}{m\|x_i - c_j\|^2} \right)^{\frac{1}{m-1}} \Rightarrow 1 = \sum_{j=1}^N \left(\frac{m\|x_i - c_j\|^2}{\lambda_i} \right)^{\frac{1}{1-m}}$$

$$\Rightarrow \lambda_i = \left(\sum_{j=1}^N \left(m\|x_i - c_j\|^2 \right)^{\frac{1}{1-m}} \right)^{1-m} \Rightarrow \mu_{ij} = \frac{1}{\sum_{k=1}^N \left(\frac{\|x_i - c_j\|^2}{\|x_i - c_k\|^2} \right)^{\frac{1}{m-1}}}$$



Q5 الگوریتم Region Growing را می‌توان به شیوه‌های مختلفی بهبود بخشید. این روش‌ها شامل راهکارهایی برای تشخیص بهتر مرزها و گسترش محدوده‌ی در حال رشد و همچنین استفاده از ویژگی‌هایی برای تضمین یکپارچگی ناحیه‌ی مدنظر می‌شوند. برای مثال می‌توان در هر مرحله، روشنایی خانه‌های واقع در مرز محدوده را به عنوان معیار بررسی خانه‌های مجاور جدید قرار داد؛ یا می‌توان از نوعی تابع خودهمبستگی برای سنجش میزان یکپارچگی محدوده‌ی انتخاب شده استفاده کرد.

Q6 روش level-set با استفاده از مجموعه‌های تراز (مجموعه ورودی‌های تابع با خروجی معین) به یافتن مرزها و محل تلاقی سطوح مختلف در تصویر می‌پردازد. این روش به فرموله کردن منحنی‌های سطوح نیاز ندارد و به صورت عددی تغییرات اشکال را بررسی می‌کند. به علاوه این روش برای تشخیص مرز در تصاویر متغیر در زمان نیز مناسب است.

الگوریتم snake از الگوریتم‌های active contour است که با داشتن اطلاعات اولیه و قیدهایی از شکل کلی تصویر، به درون‌یابی و بهبود مرحله به مرحله‌ی مرز تصویر می‌پردازد. این روش از یک اسپلاین (نوعی تابع چندجمله‌ای مقطع) شکل‌پذیر به عنوان مرز استفاده می‌کند که توسط دو تابع انرژی درونی و بیرونی کنترل می‌شود. تابع انرژی درونی روی همواری و انحنای مرز اعمال اثر می‌کند در حالی که تابع بیرونی با داشتن اطلاعات شکل، تطابق مرز بر روی تصویر اصلی را تضمین می‌کند. این الگوریتم، توسط روش کاهش گرادیان به کمینه کردن تابع انرژی کل پرداخته و مرز را به‌روز می‌کند. از مشکلات این الگوریتم می‌توان وابستگی به اطلاعات اولیه‌ی شکل، ناتوانی در تشخیص مرزهای پیچیده و حساسیت به نویز را برشمرد.

در برنامه‌ی demo ابتدا یک بیضی (قرمز) به عنوان شمای کلی شکل دور تصویر مغز قرار می‌گیرد و سپس با شروع از همین بیضی به عنوان مرز، مرحله به مرحله الگوریتم snake را اجرا کرده و مرز را بهبود می‌دهیم تا به شکل نهایی (حاوی مرز زرد) زیر برسیم.

