

باسمه تعالی



پردازش تصاویر پزشکی

بهار ۱۴۰۱-۰۲

تمرین سری چهارم

استاد: دکتر فاطمی‌زاده

علی نوریان ۹۸۱۰۲۵۲۷

۱ سوالات تئوری

۱.۱ سوال اول

۱.۱.۱

هدف این مقاله پیشنهاد یک الگوریتم جدید خوشه‌بندی تصویر MRI مغز بر اساس الگوریتم خوشه‌بندی FCM برای بهبود دقت تقسیم‌بندی است. این مقاله یک روش یادگیری multitask (چندکاره) برای FCM معرفی می‌کند که اطلاعات عمومی بین تسک‌های segmentation مختلف را پیدا می‌کند. سپس adaptive task weight learning را معرفی می‌کند که در مجموع به روش weighted multitask fuzzy C-means یا به اختصار WMT-FCM می‌انجامد. با استفاده از این روش هر تسک segmentation یک وزن بهینه‌ای پیدا می‌کند که با ترکیب با سایر تسک‌ها به یک نتیجه خوشه‌بندی بهتری می‌رسد.

۲.۱.۱

الگوریتم پیشنهادی مزیت‌ها الگوریتم FCM و روش multitask learning را با هم ترکیب می‌کند. این کار استفاده توأم از اطلاعات عمومی بین تسک‌های مختلف و اطلاعات درون هر تسک را فراهم می‌کند.

۳.۱.۱

دیتاستی که برای ارزیابی الگوریتم از آن استفاده شده است، یک دیتاست شبیه‌سازی شده از McConnell BrainWeb می‌باشد. این وبسایت مربوط به مرکز McConnell Brain Imaging Center از موسسه Montreal Neurological در دانشگاه McGill می‌باشد.

این دیتاست شبیه‌سازی شده شامل لایه‌های مختلف، ضخامت‌های مختلف، نویز لول‌های مختلف و میدان شدت سطوح غیریک‌نواخت (INC) مختلف است. دیتای استفاده شده در اینکار از ۹ دسته تصاویر T1-weighted MRI و از هر کدام لایه ۹۰ام آن استفاده شده است.

۴.۱.۱

۳ معیار مورد استفاده قرار گرفته است:

۱. Dice Similarity Coefficient (DSC)

$$DSC = \frac{2|S_1 \cap S_2|}{|S_1| + |S_2|}$$

۲. Average Dice Similarity Coefficient (DSC_av)

The average Dice similarity coefficient of WM, GM, and CSF:

$$DSC_{av} = \frac{2(|S_1^{CSF} \cap S_2^{CSF}| + |S_1^{GM} \cap S_2^{GM}| + |S_1^{WM} \cap S_2^{WM}|)}{|S_1^{CSF}| + |S_2^{CSF}| + |S_1^{GM}| + |S_2^{GM}| + |S_1^{WM}| + |S_2^{WM}|}$$

۳. Segmentation Accuracy (SA)

$$SA = \frac{\sum_{i=1}^C A_i \cap B_i}{\sum_{l=1}^C B_l}$$

۵.۱.۱

نرمال سازی: این تکنیک شدت ها را به یک محدوده پیش تعیین شده، معمولاً بین ۰ و ۱، مقیاس می دهد. کمینه ترین مقدار شدت به ۰ نگاشت می شود و بیشینه ترین مقدار شدت به ۱ نگاشت می شود، و سایر مقادیر به طور متناسب در بین آنها مقیاس می شوند.

تغییر اندازه: تغییر اندازه، یکی از تکنیک های پیش پردازش مهم که به طور معمول در تصاویر MRI استفاده می شود، است. تغییر اندازه شامل تغییر ابعاد فضایی تصویر با حفظ نسبت ابعاد آن است. این تکنیک اغلب برای استاندارد سازی ابعاد تصویر یا تطابق با نیازهای تجزیه و تحلیل یا مدل خاصی استفاده می شود.

اصلاح میدان تغییر: میدان تغییر به تغییرات فضایی آرام شدت در تصاویر MRI اشاره دارد که ممکن است به دلیل نقص ها در سیستم تصویربرداری وجود داشته باشد. تکنیک های اصلاح میدان تغییر هدف از بین بردن این نقص فضایی را دارند که می تواند دقت مراحل تجزیه و تحلیل تصویر بعدی را بهبود بخشد.

هم نمایی هیستوگرام: این تکنیک هیستوگرام یک تصویر MRI را با یک تصویر مرجع یا یک هیستوگرام هدف مطابقت می دهد. این تکنیک توزیع شدت تصویر را با توزیع تصویر مرجع هم نما می کند که می تواند برای تطبیق دادن اسکن های مختلف یا افزایش ویژگی های خاص مفید باشد.

۶.۱.۱

مقادیر بهینه پارامترهای به صورت زیر است:

$$\lambda \in \{20, 40, 60, 80, 100\}$$

$$\gamma \in \{0.2, 0.4, 0.6, 0.8, 1, 1.2\}$$

$$K = 100, \quad \text{iterations of number maximum}$$

$$\epsilon = 0.0001, \quad \text{parameter termination}$$

$$m = 2, \quad \text{index cluster}$$

۷.۱.۱

با استفاده از دو معیار SA و CSF_{av} که بالاتر اشاره شد، این روش از نظر دقت تقسیم بندی با دو روش دیگر مقایسه شده است. جدول و نمودار زیر این مقایسه را با دو روش FCM و DMFCM نشان می دهد:

TABLE 3: The average values in terms of segmentation accuracy and average Dice similarity coefficient on simulated MRI brain images.

Image	SA			DSC _{av}		
	FCM	DMFCM	WMT-FCM	FCM	DMFCM	WMT-FCM
Image 1	0.8943	0.9531	0.9870	0.8059	0.9230	0.9737
Image 2	0.8668	0.9131	0.9705	0.7425	0.8366	0.9408
Image 3	0.8289	0.8765	0.9537	0.6705	0.7716	0.9081
Image 4	0.9142	0.9630	0.9914	0.8469	0.9390	0.9826
Image 5	0.8268	0.8686	0.9543	0.6752	0.7541	0.9090
Image 6	0.7979	0.8618	0.9237	0.6107	0.7415	0.8495
Image 7	0.8781	0.9164	0.9818	0.7789	0.8521	0.9632
Image 8	0.8905	0.8865	0.9845	0.8089	0.7932	0.9688
Image 9	0.8712	0.8778	0.9737	0.7572	0.7677	0.9472

شکل ۱: جدول مقایسه روش پیشنهادی مقاله با دو روش ذکر شده دیگر

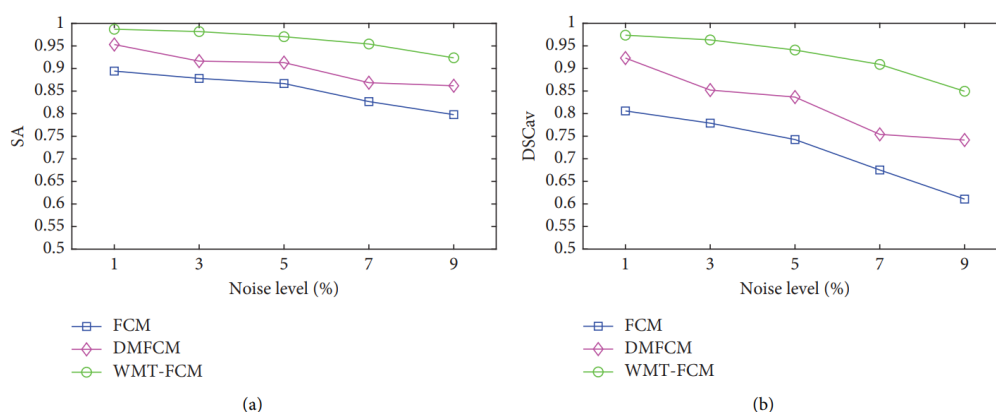


FIGURE 4: The variations of SA and average DSC on images with 20% INU and increasing noise level by FCM, DMFCM, and WMT-FCM: (a) SA and (b) DSC_{av}.

شکل ۲: نمودار مقایسه روش پیشنهادی مقاله با دو روش ذکر شده دیگر

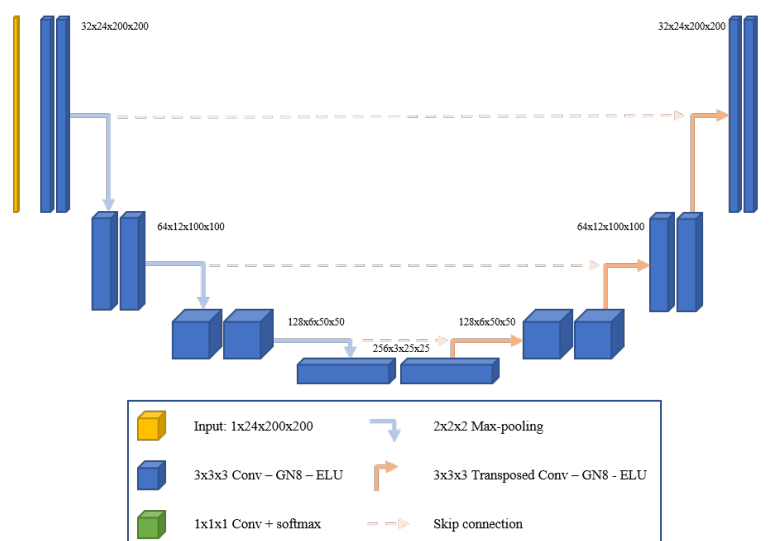
۲.۱ سوال دوم

شبکه U-Net یکی از شبکه‌های عمیق شناخته شده است که در حوزه‌های ناحیه‌بندی تصویر به خصوص در تحلیل تصاویر پزشکی مورد استفاده قرار می‌گیرد. نام این ساختار U-Net از U شکل بودن آن آمده است که شامل یک بخش encode و یک بخش decode است.

مسیر کاهش (بخش encode): سمت چپ معماری U شکل این شبکه از چندین لایه کانولوشنی و لایه‌های pooling تشکیل شده است که به صورت کلی ابعاد تصویر را کاهش می‌دهد و در عین حال عمق ویژگی‌ها را افزایش می‌دهد. این مسیر اطلاعات زمینه‌ای سطوح بالا را بدست می‌آورد.

مسیر افزایشی (decoder): سمت راست این شبکه مسیر افزایشی است که با استفاده از transpose convolutions (که با نام‌های upsampling یا deconvolution نیز شناخته می‌شود) و اضافه کردن ویژگی‌هایی در بایه متناظر در مسیر کاهش، ابعاد تصویر را افزایش می‌دهد. اضافه کردن ویژگی‌های مسیر کاهش در این مسیر کمک می‌کند اطلاعاتی مکانی که در مسیر کاهش از دست رفته بود بازیابی شوند.

خروجی این شبکه یک ماتریس ناحیه‌بندی است که مشخص می‌کند هر پیکسل به کدام کلاس، دسته یا ناحیه تصویر ورودی مربوط می‌شود.



شکل ۳: معماری شبکه U-Net

الگوریتم آموزش شبکه U-Net معمولاً شامل مراحل زیر است:

تهیه مجموعه داده: در این مرحله، باید مجموعه داده‌ای که شامل تصاویر اصلی و ماسک‌های متناظر با تقسیم‌بندی است، آماده شود. تصاویر باید به همراه ماسک‌های صحیح برچسب‌گذاری شده باشند.

تعیین تابع هدف: برای آموزش شبکه U-Net، باید تابع هدف یا تابع خطا مناسبی را تعیین کنید. معیارهایی مانند تابع تلفات چندجمله‌ای یا تابع تلفات متقاطع معمولاً برای مسائل segmentation استفاده می‌شوند.

ساختار مدل: باید ساختار شبکه U-Net را با استفاده از لایه‌های کانولوشنی، لایه‌های upsampling و کانکشن‌های انتقال (skip connections) ایجاد کنید. این مدل معمولاً با استفاده از یک شبکه عصبی کانولوشنی عمیق قبل آموزش (pre-trained) مانند شبکه‌های VGG یا ResNet نیز ترکیب می‌شود.

تعیین الگوریتم بهینه‌سازی: شما باید یک الگوریتم بهینه‌سازی مانند Adam یا SGD را برای آموزش شبکه انتخاب کنید. همچنین، پارامترهای بهینه‌سازی نیز باید تنظیم شوند، از جمله نرخ یادگیری (learning rate) و معیارهای دیگری که توسط الگوریتم بهینه‌سازی استفاده می‌شود.

آموزش مدل: در این مرحله، مدل U-Net با استفاده از مجموعه داده آموزشی آموزش داده می‌شود. تصاویر و ماسک‌های متناظر به عنوان ورودی به شبکه داده می‌شوند و خروجی شبکه با ماسک‌های واقعی مقایسه می‌شود. سپس تابع خطا محاسبه شده و پارامترهای شبکه با استفاده از الگوریتم بهینه‌سازی به‌روزرسانی می‌شوند. این عملیات آموزش بر روی دسته‌های تصاویر تکرار می‌شود تا بهبودی پیدا کند.

ارزیابی و تست: پس از آموزش، مدل بر روی مجموعه داده ارزیابی می‌شود. معیارهایی مانند دقت (accuracy)، صحت (precision)، بازیابی (recall) و اندازه‌گیری‌های دیگر می‌توانند استفاده شوند. همچنین، می‌توانید مدل را بر روی تصاویر جدیدی تست کنید تا نتایج تقسیم‌بندی را بررسی کنید.

با تکرار مراحل آموزش، بهبود مدل U-Net در تقسیم‌بندی تصاویر را خواهید داشت.

۲ سوالات عملی

۱.۲ سوال اول

موجود در فایل Jupyter notebook با نام MIP_HW4_Q1_98102527.

۲.۲ سوال دوم

موجود در فایل Jupyter notebook با نام MIP_HW4_Q2_98102527.

۳.۲ سوال سوم

۱.۳.۲ بخش اول

موجود در فایل Jupyter notebook با نام MIP_HW4_Q3_part1_98102527.

۲.۳.۲ بخش دوم

ابتدا به صورت زیر داده را لود کرده و تصویر آن را ذخیره می‌کنیم:

```
load 'data.mat';

scaledData = mat2gray(imageData);
imwrite(scaledData, 'imageData.jpg');

scaledData = mat2gray(imageMask);
imwrite(scaledData, 'imageMask.jpg');
```

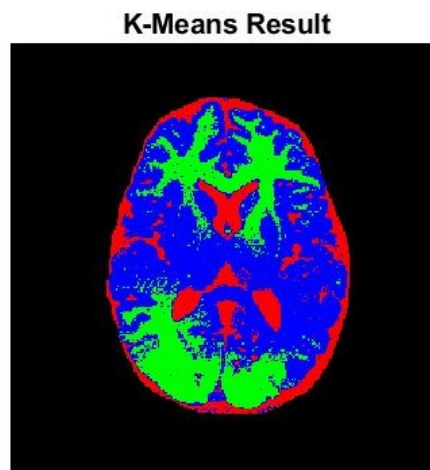
سپس با توجه به مقادیر ذکر شده در دستور کار الگوریتم K-Means را روی تصویر داده شده اجرا می‌کنیم:

```
mask = single(imageMask);
img = single(imageData);
[R, C] = size(mask);

k = 3;
q = 5;
b_const = 1;
b_init = (b_const * ones(R, C)) .* mask;
f = fspecial('gaussian', 9, 2.5);
eps = 1e-7;
[seg, mu, sigma] = KMeans(img, mask, k, eps);

path = 'res_kmeans';
title_str = 'K-Means Result';
showSegmented(seg, k, title_str, path);
```

نتیجه این الگوریتم به صورت زیر است:



شکل ۴: خروجی الگوریتم K-Means

اکنون می‌خواهیم از الگوریتم داده شده توسط سوال استفاده کنیم. ابتدا مقادیر اولیه متغیرها را با توجه به مقادیر گفته شده و همچنین مقادیر بدست آمده از روش K-Means، مقداردهی می‌کنیم.

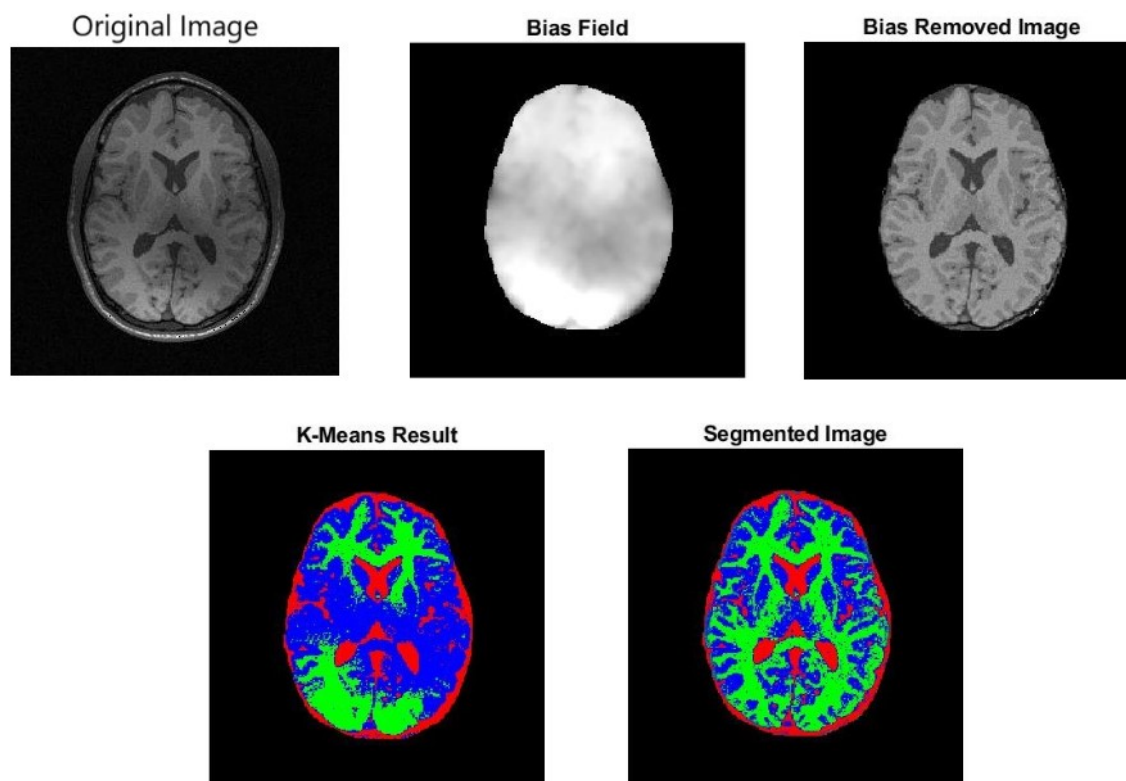
مقادیر اولیه متغیرهای ورودی تابع را به صورت زیر مقداردهی و الگوریتم را اجرا می‌کنیم:

```
u = zeros(R, C, k);           % The class memberships
for i=1:k
    u(:, :, i) = (seg == i);
end
b = b_init;                   % The bias field
c = mu;                       % The class means
q = 5;                        % The q-parameter as specified in the slides
w = f;                        % The neighbourhood mask
J_init = objectiveFunction(img, b, c, q, u, w);
eps = 1e-5;
N_max = 250;
[u, b, c, J] = iterate(img, mask, u, b, c, q, w, J_init, eps, N_max);
```

با توجه به بدست آوردن ماتریس membership و bias field، تصویر bias removed را به صورت زیر بدست می‌آوریم.

```
bias_removed_image = mask .* img ./ b;
```

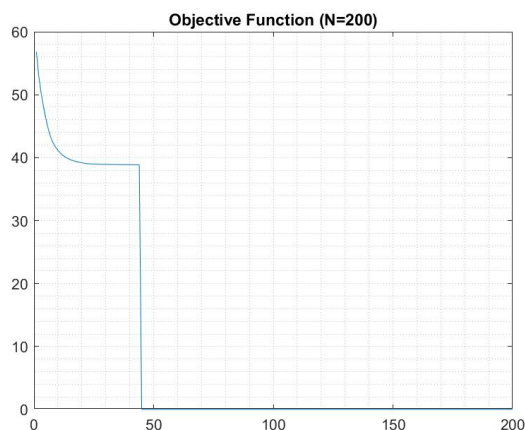
نتایج بدست آمده از این الگوریتم به صورت زیر است:



شکل ۵: تصویر اصلی، خروجی‌های الگوریتم و خروجی K-Means

همانطور که مشاهده می‌شود نتیجه بدست آمده بسیار بهتر از نتیجه K-Means است. همچنین bias field تصویر نیز به خوبی تشخیص داده شده است و با حذف آن از تصویر، تصویر واضح‌تر و باکیفیت‌تری بدست آمده است.

همچنین مقدار تابع هدف را نیز به ازای تکرارهای مختلف (تا ۲۰۰ تکرار) رسم می‌کنیم. (مقادیر تابع هدف به صورت یک بردار در خروجی تابع الگوریتم بدست می‌آید). لازم به ذکر است با توجه به مقدار آستانه الگوریتم در ایتريشن ۴۴ متوقف می‌شود.



شکل ۶: مقادیر تابع هدف تا ۲۰۰ تکرار با آستانه $1e-5$

تمامی تصاویر و نتایج به صورت جداگانه همراه گزارش ارسال شده است. در گزارش موارد مهم آورده شده است.