



## تشخیص ام اس از روی تصاویر ام آر آی

محمدرضا ایروانی<sup>۱</sup>، نساء شریفی<sup>۲</sup>، کیان شهرياری<sup>۳</sup> و صدف محرری<sup>۴</sup>\*

<sup>۱</sup>دانشگاه آزاد اسلامی واحد خمینی شهر، mreza.iravani.ai@gmail.com

<sup>۲</sup>دانشگاه آزاد اسلامی واحد خمینی شهر، nesa.sharifi74@gmail.com

<sup>۳</sup>دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران، kian.shahriari@gmail.com

<sup>۴</sup>دانشگاه آزاد اسلامی واحد خمینی شهر، sadaf.moharreri@gmail.com

\*سرپرست تیم

چکیده - بیماری ام اس یک بیماری عصبی فلج کننده مغز و نخاع است که اغلب منجر به زوال شناختی و ناتوانی جسمی می شود. ایجاد یک روش سریع و مناسب برای تخمین تعداد و اندازه ی ضایعات MS در مغز، برای تخمین پیشرفت بیماری و اثر بخشی درمان بسیار مهم است. اما، با توجه به تغییرات مکرر ضایعات در موقعیت مکانی، اندازه، تنوع مورفولوژی (ریخت شناسی)، شباهت شدت آنها با بافت طبیعی مغز و تغییرات آناتومیک موارد داخل تصاویر مغزی؛ شناسایی درست، توصیف و تعریف مشخصات ضایعات MS در تصویر رزونانس مغناطیسی (MRI) به شدت سخت است. برای تشخیص قسمت های درگیر بیماری MS در این چالش، با بهره گیری از الگوریتم کانولوشن سه بعدی قادر به ساخت مدلی با قابلیت تشخیص پیکسل های درگیر بیماری به همراه شماره منطقه مربوطه باشد. مدل در نهایت قادر به پیش بینی ماسک پیش بینی شده با نرخ DSC ۳۵٪ را دارد و در نهایت خروجی تصاویر داده های تست را ارائه می دهد.

کلید واژه- MS، شبکه عصبی کانولوشن، پردازش تصویر، FCN.

بیماری MS از روی تصاویر MRI مغزی افراد شود. این مدل با بهره گیری از عملیات کانولوشن سه بعدی (3D Convolution) قادر به تولید خروجی مورد نظر می باشد.

### ۱- مقدمه

ام اس یا Multiple sclerosis (MS) یک بیماری در سیستم عصبی مرکزی (CNS) است و به غلاف میلین که مانند عایقی اطراف آکسون ها در مغز است، آسیب می رساند. این بیماری باعث می شود که سیستم ایمنی به فیبرهای عصبی حمله کند. تا به امروز هیچ نشانگر زیستی تصویربرداری قطعی برای تشخیص ام اس وجود ندارد [۱]. معیارهای تشخیصی فعلی عمدتاً مبتنی بر عودهای بالینی است که با وجود ضایعات ماده سفید (WMLs) در MRI پشتیبانی می شود [۲]. با این حال تصویر برداری MRI به دلیل دقت مناسب در تشخیص بافت های نرم و نمایش تمایز بافت های طبیعی از غیر طبیعی جز ابزارهای مهم در عرصه تشخیص انواع بیماری ها به خصوص بیماری های مغزی می باشد. تصویر برداری MRI نشان داده است که نسبت به این بیماری بسیار حساس می باشد و توانایی این را دارد که ضایعات MS را نسبت به بافت های محیطی خود با شدت متفاوتی نمایش دهد. ارزیابی دقیق از هر ضایعه MS در تصاویر MR به صورت دستی برای متخصصین پزشکی یک کار پر زحمت و انجام تقسیم بندی به صورت ذهنی بسیار پرخطا می باشد [۳، ۴]. یک جایگزین جذاب و دقیق برای تقسیم بندی دستی، تقسیم بندی کامپیوتری می باشد [۵] که می تواند با دقت بیشتری و در مدت زمان کمتری این کار را برای پزشک انجام دهد. در کارهای اخیر که انجام شده است، محققین تمرکز زیادی بر بهبود تقسیم بندی ضایعات مغز و به خصوص ضایعات حاصل از بیماری ام اس که کاری نسبتاً مشکل می باشد پرداخته اند که از الگوریتم های مختلفی برای این کار استفاده کرده اند که از جمله آنها می توان به روش graph cuts که در مقالات مختلف از جمله [۶] استفاده شده است اشاره کرد که با همراه کردن مارکوف مدل [۷] MRF توانسته است نتایج نسبتاً خوبی بدست آورد ولی در تعیین لبه ها دچار مشکل می باشند و به همین دلیل از Edge based segmentation نیز برای رفع مشکل و تقسیم بندی بر اساس لبه ها استفاده می کنند ولی در نواحی مختلف امکان نادیده گرفته شدن و عدم برچسب گذاری وجود دارد به همین دلیل از فیلتر های همچون گابور برای آماده سازی در ابتدای پردازش استفاده می کنند [۸].

### ۲- داده ها

به منظور شرکت در این مسابقه ۷۵ تصویر MRI از بیماران ام اس در اختیار تیم قرار گرفت. تصویربرداری با دستگاه MRI (Siemens company, Germany, 1.5 Tesla MRI Siemens Avanto scanner system Siemens, Henkestr Erlangen) انجام شده است. بیمار به حالت خوابیده روی تخت قرار گرفته و سر بیمار برای تصویربرداری در یک سیم پیچ دوازده کانالی قرار گرفته و تصویر تهیه شده است. تصاویر به دست آمده دارای فرمت NIFTI و ابعاد ۲۳۶ در ۲۵۶ در ۱۶۰ می باشند. جهت سهولت استفاده و رفع ایراد تغییراتی در داده ها اعمال شد. توضیحات تغییرات در قسمت بعد به تفصیل بیان شده است.

در مرحله اول مجموعه داده آموزش شامل ۵۵ تصویر و در مرحله دوم ۲۰ داده در اختیار گروه قرار گرفت و تصاویر ماسک از آنها دریافت شد.

### ۳- روش کار

در ابتدا لازم به ذکر است، زبان استفاده شده در این پروژه پایتون (Python) و کتابخانه اصلی دیپ لرنینگ مورد استفاده تانسورفلو (TensorFlow) [۹] می باشد. به منظور ایجاد محیط تعاملی تر و قابل درک به همراه توضیحات و کامنت های مربوط به هر کد، نوت بوک جویپتر (Jupyter) با پسوند فایل "ipynb" انتخاب شده است که امکان اجرای کد ها را بصورت قطعه قطعه ارائه می دهد. به دلیل مشکل بهم ریختگی زبان فارسی در بعضی از نسخه های جویپتر، روش کار و توضیحات بصورت خیلی جامع و دقیق بصورت انگلیسی در کنار کد ارائه شده است.

الگوریتم مورد استفاده در این چالش برگرفته از شبکه UNet می باشد [۱۰] که از شبکه عصبی تمام کانولوشنی ۳ بعدی Fully



فرد سالم به تعداد ۱۰ بار و دو داده دارای همه لیبیل‌ها بخصوص بخاطر کم بودن داده دارای لیبیل ۵، هرکدام ۴ مرتبه تکرار شدند و تعداد داده‌های در دسترس برای آموزش به ۵۸ عدد داده تبدیل می‌شود.

### ۱-۳- شبکه عصبی کانولوشن

برای آموزش دادن شبکه به دلیل سنگین بودن شبکه و بزرگی اندازه تصاویر و محدودیت منابع محاسباتی، از Google Collab با کارت گرافیک Nvidia T4 با محدودیت حداکثر ۴ ساعته در روز استفاده شده است. این محدودیت قدرت محاسباتی اجازه تقویت مدل را برای دریافت نتایج بهتر مسیر نمی‌کند و در همین حین بارها دچار عدم اجازه دسترسی به طور موقت از سوی گوگل بابت آموزش دادن شبکه شدیم. دلیل پرش ناگهانی تعداد فیلترها از ۱۶ به ۱۲۸ نیز بخاطر محدودیت قدرت محاسباتی با در نظر گرفتن بهترین شبکه ممکن بود.

به دلیل بزرگ بودن سایز تصاویر، تصاویر بصورت جنراتوری (generator) به شبکه داده شده‌اند که با تابع `generator` در پروژه معرفی می‌شود. سپس داده‌ها آموزش، به وسیله تابع `train_test_split` کتابخانه `scikit-learn` [۱۱] خود به دو بخش آموزش و تست جدا با نسبت ۸۰٪ داده آموزش و ۲۰٪ داده تست که این داده تست خود بعنوان `validation set` استفاده می‌شود، تقسیم شدند. بعد از تقسیم بندی ۵۸ عدد داده به قسمت آموزش و ۱۵ عدد به قسمت تست اختصاص داده شدند. سپس این تقسیم بندی با نمونه سازی از جنراتور ساخته شده به دو جنراتور `train_datagen` برای ساختن داده آموزش و `test_datagen` برای ساختن داده تست ساخته شدند. شبکه مورد استفاده با بهره گیری از `3D Convolution` ابتدا تصویر را دریافت می‌کند و سپس لایه خروجی با ۱۶ فیلتر با سایز کرنل (۳،۳) و به دنبال آن لایه `DMaxPool` با استراید (۲،۲) را خروجی میدهد. معماری شبکه مورد استفاده در جدول ۲ برای بررسی بیشتر قرار داده شده است. معیار ارزیابی درست ماسک‌های پیش بینی شده با ماسک مرجع `Dice Coefficient(DSC)` نام دارد که بسیار مقداری نزدیک و مشابه به `Intersection Over Union(IoU)` دارد و رابطه آن به شرح زیر است:

$$DSC(A, B) = \frac{2|X \cap Y|}{|X| + |Y|} \quad (3)$$

تابع ضرر مورد استفاده نیز تابع `Dice Loss` می‌باشد (معادله ۴) که در کد‌ها با نام `soft_dice_loss` شناخته می‌شود و رابطه آن به شرح زیر است:

$$\mathcal{L}_{Dice}(p, q) = 1 - \frac{2 \times \sum_{ij} p_{ij} q_{ij} + \epsilon}{(\sum_{ij} p_{ij}^2) + (\sum_{ij} q_{ij}^2) + \epsilon} \quad (4)$$

برای رفع مشکلات حساسیت شبکه به سایز تصاویر ورودی، سایز معادل ورودی آنها برابر `None` در نظر گرفته شده است. با وجود سایز `None`، برای پیش بینی کردن نتیجه، `step` مورد نظر باید بر روی ۱ قرار بگیرد که پس از پیش بینی توسط شبکه سایز معادل تصویر ورودی را بگیرد.

### ۴- نتایج

بعد از ساخت شبکه مورد استفاده، نوبت به کامپایل آن می‌رسد. تابع بهینه ساز شبکه `Adam` با نرخ یادگیری `1e-4` می‌باشد. مدل به مدت ۴ ساعت با ۴۰ اپیک (Epoch) آموزش داده شده است و در نهایت به مقدار `loss` ۰/۵۳۵۸ و مقدار `DSC` ۰/۳۵۵۵ برای دیتای ارزیابی (Validation) شده است.

پس از آموزش دادن مدل، برای درک بهتر از عملکرد آن، نسبت به داده‌های ارزیابی که داشتیم یک پیش بینی برای داده شماره ۱۹ توسط مدل گرفتیم. نتیجه این پیش بینی در تصویر ۲ و ۳ برای بررسی بیشتر قرار گرفته است.

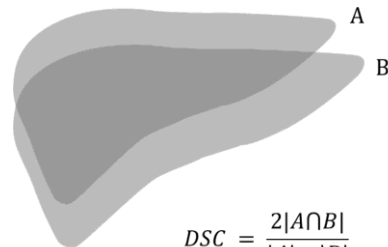
`Convolutional Network (FCN)` استفاده شده است. شبکه دارای دو مرحله پایین رو (`Contracting Path`)، یک مرحله در قسمت `Bottom Layer` و دو مرحله بالارو (`Expanding Path`) می‌باشد. دلیل تعداد کم مراحل پایین رو و بالارو، ناهمگنی اندازه تصاویر و ماسک‌های ارسالی برای چالش است. برخلاف وبسایت مرجع که سایز تصاویر را (۲۳۶،۲۵۶،۱۶۰) اعلام کرده بود، تعداد کمی از تصاویر فقط دارای این سایز بودن و تصاویر دارای سایزهای مختلفی اعم از (۲۲۴،۲۵۶،۱۶۰)، (۲۵۶،۲۵۶،۱۷۶) بودند که خود مشکل بسیار بزرگی برای پردازش داده ایجاد می‌کند. در نهایت بدون در نظر گرفتن حساسیت شبکه به سایز ورودی و با اختصاص دادن "None" به سایز ورودی، با در نظر گرفتن تعداد مسیرهای پایین رونده قابل تقسیم بر سایز تصاویر ورودی مشکل سایز ورودی برطرف شد. اما عدم هماهنگی سایز می‌تواند تاثیر چشم گیری بر جلوگیری از پیشرفت شبکه بگذارد. در این پژوهش ابتدا ماسک‌های داده‌ها را بصورت به شبکه دادیم ولی مدل مورد استفاده قادر به پردازش داده‌های بصورت غیر از ۰ و ۱ نبود و دقت مدل از رضایت کافی برخوردار نبود. برای حل این مشکل، تصاویر ماسک را مثل یک نقشه توپوگرافیک بررسی کردیم. به وسیله آنالیز داده‌ها مشخص شد که لیبیل‌های ماسک از ۱ تا ۵ متغیر هستند و همه تصاویر الزاما لیبیل‌های پشت سر هم نداشتند. به طور مثال، برخی از داده‌ها دربرگیرنده تمامی لیبیل‌ها و بعضی فقط دارای لیبیل ۱ و ۲. برای رفع این مشکل یک محور دیگر به تنسور تصویر اضافه شد. ایم محور دارای سایز ۵ است که برگرفته از ۵ کلاس مختلف در حالت حداکثر می‌باشد و به ازای هر وجود پیکسل کلاس مرتبط، در آن محور ۱ جایگذاری شد. با این روش در هر لایه دو بعدی، لایه سوم تازه اضافه شده شبیه یک لایه توپوگرافیک ۳ بعدی دیده می‌شود و برای مدل، پردازش امکان پذیر می‌شود. برای درک بهتر، اگر سایز ماسک مورد نظر ما (۲۳۶،۲۵۶،۱۶۰،۵) باشد، به (۲۳۶،۲۵۶،۱۶۰،۵) تبدیل می‌شود. این آماده سازی در پروژه به وسیله تابع `custom encode` انجام شده است.

پس از این مرحله، داده‌های ورودی به وسیله تابع `standardize` نرمالیزه می‌شوند و این عمل باعث می‌شود مدل سریعتر و با پردازش محاسباتی کمتر تبدیل شود. این تابع در هر محور `z` با شماره محور ۲ پیمایش می‌کند و تابع زیر را اجرا میکند.

$$Centered = image - mean(image) \quad (1)$$

$$Normalized\ image = \frac{Centered}{Std} \quad if\ std \neq 0 \quad (2)$$

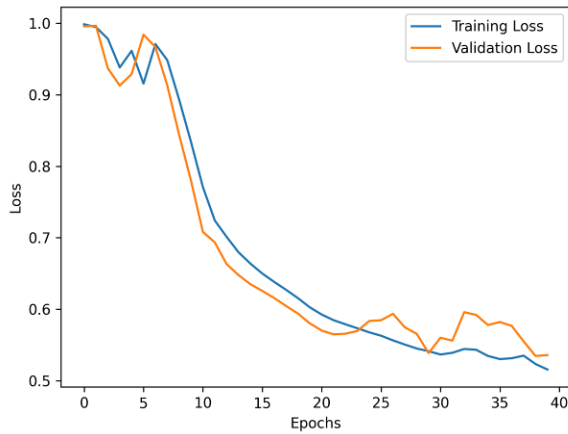
تصویر ۱ برای ایجاد درک بهتر نسبت به معیار اندازه گیری `DSC` ارائه شده است.



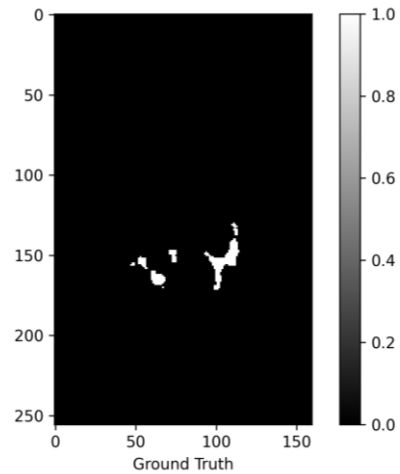
DSC: Dice similarity coefficient

تصویر ۱. معیار DSC

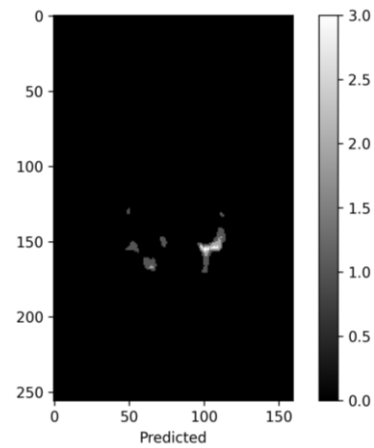
بعد از بررسی‌های انجام شده در مورد در تعادل بودن لیبیل داده‌ها، دریافتیم داده‌هایی مانند تصویر فرد سالم کمتر در دیتاست موجود هستند. بدین منظور افزایش داده (Oversampling) دستی صورت گرفت و داده



تصویر ۵. معیار Loss در طول آموزش مدل



تصویر ۲. ماسک واقعی



تصویر ۳. ماسک پیش‌بینی شده توسط مدل

در نهایت، با استفاده از داده‌های تست، ماسک مورد نظر برای تصاویر تست چالش پیش‌بینی شده و بعد از اعمال آستانه (threshold) ۰/۵، داده‌های زیر آستانه به ۰ و بالای آستانه به ۱ تبدیل شدند. سپس در محور آخر با یکدیگر جمع شدند و ماسک نهایی بازسازی شد. فایل‌های بدست آمده با فرمت "nii" و با الگوی اسم "Mask\_X.nii" ذخیره شدند.

جدول ۲. معماری شبکه

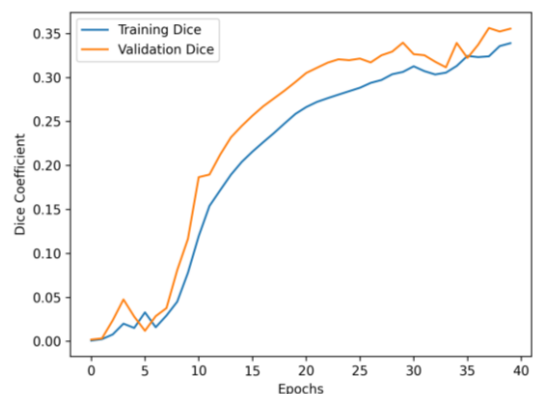
Layer	Input	Filters	Output
Conv1	(N*, N, N)	16	(N, N, N, 16)
Conv2	(N, N, N, 16)	16	(N, N, N, 16)
Max1	(N, N, N, 16)		(N, N, N, 16)
Conv3	(N, N, N, 16)	128	(N, N, N, 128)
Conv4	(N, N, N, 128)	128	(N, N, N, 128)
Max2	(N, N, N, 128)		(N, N, N, 128)
Conv5	(N, N, N, 128)	256	(N, N, N, 256)
Up1	(N, N, N, 256)	128	(N, N, N, 128)
Concat1 (Up1, Conv4)	(N, N, N, 128) (N, N, N, 128)		(N, N, N, 256)
Conv6	(N, N, N, 256)	128	(N, N, N, 128)
Conv7	(N, N, N, 128)	128	(N, N, N, 128)
Up2	(N, N, N, 128)	16	(N, N, N, 16)
Concat2 (Up2, Conv2)	(N, N, N, 16) (N, N, N, 16)		(N, N, N, 32)
Conv8	(N, N, N, 32)	16	(N, N, N, 16)
Conv9	(N, N, N, 16)	16	(N, N, N, 16)
Output	(N, N, N, 16)	5	(N, N, N, 5)

\* N: None

در نگاه اول ممکن است بنظر برسد مدل دقت مناسبی ندارد اما اگر به مقیاس بندی روشن بودن در تصویر ۳ دقت کنیم، متوجه می‌شویم که ماسک دارای شماره ۱ تاریک تر بنظر می‌رسد و از نظر محل تشخیص بسیار شبیه ماسک واقعی داده‌های ارزیابی است. معیارهای سنجش دقت مدل در طول یادگیری بر روی داده آموزش و داده ارزیابی در جدول ۱، تصویر ۵ و ۶ برای بررسی بیشتر ارائه شده است.

جدول ۱. معیارهای ارزیابی مدل

Epochs	Loss	DSC	Val_Loss	Val_DSC
40	0.5157	0.3390	0.5358	0.3555



تصویر ۴. معیار DSC در طول آموزش مدل



## ۵- نتیجه گیری

در این مطالعه با در نظر گرفتن عدم هماهنگی سایز تصاویر ورودی MRI مغز برای تشخیص MS، مدلی ارائه شده که حساسیتی نسبت به این موضوع نداشته نباشد. گرچه این موضوع باعث کاهش دقت مدل در ارائه پیش بینی های دقیق تر می شود. با توجه به توانایی اکثر مدل ها نسبت به تشخیص فقط یک کلاس در پردازش تصویر، این مدل قادر به پردازش تمامی کلاس های تعریف شده با بهره گیری از رویکرد ریاضی محور می باشد. در نهایت مدل با بهره گیری از شبکه عصبی کانولوشن ۳ بعدی، قادر به تشخیص پیکسل های درگیر بیماری با کلاس مرتبط با معیار DSC معادل ۰/۳۵۵۵ بر روی داده های ارزیابی (Validation) می باشد. با توجه به محدودیت منابع محاسباتی، مدل به خوبی قادر به تشخیص ماسک درگیر بیماری بصورت کلی نگری (Generalized) می باشد.

## منابع

- MRIs," Open Journal of Medical Imaging, vol. 1, no. 02, pp. 26-42, 2011.
- [5] B. A. Abdullah, A. A. Younis, and N. M. John, "Multi-sectional views textural based svm for ms lesion segmentation in multi-channels mris," The open biomedical engineering journal, vol. 6, p. 56, 2012.
- [6] M. S. Khadem, "MRI Brain image segmentation using graph cuts," 2010.
- [7] R. Khayati, M. Vafadust, F. Towhidkhah, and M. Nabavi, "Fully automatic segmentation of multiple sclerosis lesions in brain MR FLAIR images using adaptive mixtures method and Markov random field model," Computers in biology and medicine, vol. 38, no. 3, pp. 379-390, 2008.
- [8] A. Bijar, R. Khayati, and A. Penalver Benavent, "Increasing the contrast of the brain MR FLAIR images using fuzzy membership functions and structural similarity indices in order to segment MS lesions," Plos one, vol. 8, no. 6, p. e65469, 2013.
- [9] M. Abadi, A. Agarwal, et.al, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, 2015: Springer, pp. 234-241.
- [11] F. Pedregosa, G. Pedregosa, et.al, Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830, 2011.
- [1] A. I. AITokhis, A. M. AlOtaibi, G. A. Felmban, C. S. Constantinescu, and N. Evangelou, "Iron rims as an imaging biomarker in MS: A systematic mapping review," Diagnostics, vol. 10, no. 11, p. 968, 2020.
- [2] S. Chawla et al., "Longitudinal study of multiple sclerosis lesions using ultra-high field (7T) multiparametric MR imaging," PLoS One, vol. 13, no. 9, p. e0202918, 2018.
- [3] L. Mozaffari, A. Broumandnia, and E. S. Shahraki, "Diagnosis of MS Lesions on MRI Images."
- [4] B. A. Abdullah, A. A. Younis, P. M. Pattany, and E. Saraf-Lavi, "Textural based SVM for MS lesion segmentation in FLAIR