گزارش پروژه ۵ هوش مصنوعی

حسام رمضانیان

۸۱۰۱۰۰۲۴۸

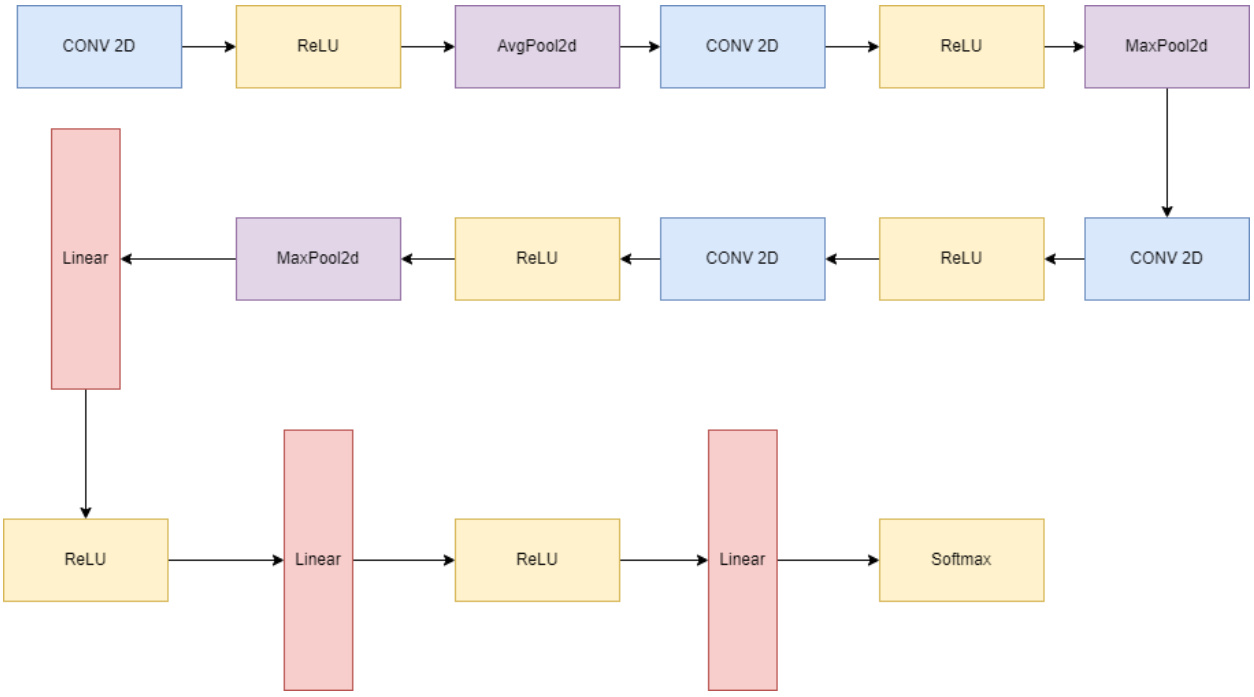
پیش پردازش :

نرمالایز کردن مقادیر موجود را در بازه -۱ تا ۱ قرار میدهد که از باعث جلوگیری از overfittingو همچنین باعث میشود که مقادیر بسیار زیاد یا کم تاثیر شدیدی بر مدل نداشته باشند اما باعث افزایش هزینه محاسباتی میشود و برخی داده ها را از دست میدهیم

نسبت داده ها :

معمولا داده های تست ۲۰ تا ۳۰ درصد داده ها را تشکیل میدهد و سایر ان برای ترین استفاده میشود تنظیم صحیح این مقادیر باعث جلوگیری از over/under fittingمیشود

معماری شبکه CNN :



برای ایجاد این معماری از معماری alexnet کمک گرفتیم یعنی مشابه این معماری در هر بخش یک لایه Conv2D یک ReLu و پس از آن یک لایه Pooling قرار دارد همانطور که در شکل مشاهده میکنید. در انتها بخش اول از AvgPool استفاده شده با اندازه کرنل ۶ دلیل این انتخاب این است که به طور معمول در alexnet کرنل ها maxpoolهستند با سایز ۳ اما در alexnet سایز ورودی ۲۲۷ میباشد که تقریبا به اندازه نصف عکس با سایز ۵۱۲ میباشد و باتوجه به اینکه میخواهیم از این pool به عنوان یک تغییر سایز دهنده استفاده کنیم از AvgPool بجای Maxpool استفاده کردیم سایز کرنل : ۳*۲ = ۶ همچنین دلیل استفاده از ReLu این است که سایر توابع فعالسازی نیاز به محسبات بیشتری دارند اما نتیجه تفاوت زیادی ندارد

در بخش بعد علت استفاده از Maxpool این است که فیچر های برجسته تر به لایه بعد انتقال یابند

در بخش بعد از ۲ لایه conv2dاستفاده کردیم به همراه ReLuتا تعداد کانال هایی که در هر مرحله افزایش دادیم را کمی کاهش داده تا در بخش تماما متصل با مشکل مواجه نشویم سپس دوباره با یک Maxpool ابعاد را کاهش دادیم و فیچر های برجسته تر را به مرحله بعد فرستادیم

در بخش نهایی سه لایه خطی قرار دادیم تا به صورت پیمانه ای تعداد نرون ها را کاهش دهد تا به تعداد کلاس ها برسیم یک لایه softmax نیز موجود است که تابع loss موجود ان را اعمال میکند

دلیل استفاده از پارامتر inplace در ReLU به این منظور است که درهمان فضا اعمال شود و فضای رم جدید اشغال نکند

انواع توابع فعالسازی :

۱. ReLU (Rectified Linear Unit): یکی از پرکاربردترین توابع فعال‌سازی است که برای هر ورودی مثبت خروجی همان ورودی و برای ورودی‌های منفی خروجی صفر است. معمولاً بعد از لایه‌های کانولوشن و در لایه‌های کاملاً متصل (Fully-Connected) مورد استفاده قرار می‌گیرد تا اثر غیرخطی مفیدی بر ویژگی‌ها اعمال کند.

۲. Leaky ReLU: نوعی اصلاح‌شده از ReLU است که به جای خروجی صفر برای ورودی‌های منفی، یک مقدار کوچک مثبت را برمی‌گرداند..در اکثر موارد می تواند جایگزین ReLU شود.

۳. Sigmoid: تابعی S-شکل است که خروجی بین ۰ و ۱ ایجاد می‌کند. در گذشته بسیار محبوب بوده اما امروزه کمتر مورد استفاده قرار می‌گیرد. برای طبقه بندی در بعضی لایه های خروجی استفاده می شود.

۴. Tanh: تابعی شبیه Sigmoid است با این تفاوت که خروجی بین -۱ و ۱ تولید می‌کند. تقریبا مزایا و معایب مشابه Sigmoid دارد.

۵. Softmax: برای طبقه‌بندی چندکلاسه به کار می‌رود و یک توزیع احتمال برای هر کلاس خروجی می‌دهد. معمولا در لایه آخر مورد استفاده قرار می‌گیرد.

تابع هزینه:

Hinge Loss:

تابع تلفات Hinge یک تابع تلفات محبوب برای مسائل طبقه‌بندی است که به جای تمرکز مستقیم بر روی احتمال‌ها، بر روی فاصله پیش‌بینی از برچسب واقعی تمرکز می‌کند. این تابع تلفات، پیش‌بینی‌هایی را که فاصلهی زیادی از برچسب صحیح دارند جریمه می‌کند. با این حال، یک حاشیه یا مارجین برای مقدار کمی انحراف در نظر گرفته می‌شود.

Cross-Entropy Loss:

تابع هزینه cross entropy به عنوان یکی از پرکاربردترین توابع هزینه در مسائل طبقه‌بندی، فاصله بین توزیع احتمال کلاس‌های پیش‌بینی شده توسط مدل و توزیع احتمال واقعی کلاس‌ها را اندازه‌گیری می‌کند. هدف از کمینه کردن این تابع هزینه افزایش احتمال متعلق به کلاس درست برای هر نمونه است. از آنجایی‌که تابع cross entropy مستقیماً بر روی توزیع احتمال‌ها عمل می‌کند، بیشتر در مسائل دسته‌بندی چندکلاسه مورد استفاده قرار می‌گیرد.

MSE:

معمولا در رگرسیون ها مورد استفاده قرار میگیرد . مقادیر پیش‌بینی شده و واقعی را مقایسه می‌کند، میانگین مربع اختلاف را می‌گیرد. کمینه کردن MSE باعث نزدیک شدن پیش‌بینی‌ها به مقادیر واقعی می‌شود.

Cosine Similarity Loss:

برای بردار هایی با ابعاد بالا مورد استفاده قرار میگیرد . به صورتی که کسینوس زاویه بین پیش بینی ها و مقادیر واقعی را اندازه گیری می کند. زاویه کوچکتر نشان دهنده تطابق بهتر بین بردارها است.

Contrastive Loss:

برای آموزش بردارهای نمایشی استفاده می‌شود که در آن هدف اصلی این است که بردارهای مربوط به نمونه‌های مشابه را به هم نزدیک‌تر و بردارهای مربوط به نمونه‌های متفاوت را از هم دور کنیم.

دلیل انتخاب Cross-Entropy :

دلیل اصلی انتخاب تابع هزینه Cross-Entropy برای این پروژه طبقه‌بندی تصاویر، کاربرد گسترده‌ی آن در شبکه‌های عصبی کانولوشنی و مسائل طبقه‌بندی چندکلاسه است.Cross-Entropy یک انتخاب متداول برای CNNهای طبقه‌بند تصاویر است زیرا این تابع، فاصله‌ی بین توزیع احتمال کلاس‌های پیش‌بینی‌شده و توزیع واقعی آن‌ها را محاسبه می‌کند. کمینه کردن این تابع تلفات باعث می‌شود تا احتمال متعلق بودن به کلاس درست، بیشینه شود.

بهینه ساز:

Adam : یک الگوریتم بهینه‌سازی آداپتیو (سازگارشونده) است که به طور خودکار نرخ یادگیری را برای هر پارامتر شبکه عصبی تنظیم می‌کند.کارکرد آن به این صورت است که برای هر پارامتر، میانگین و واریانس گرادیان‌ها در طول زمان را محاسبه می‌کند. سپس با استفاده از این آماره‌ها، نرخ یادگیری هر پارامتر را به‌روزرسانی می‌کند. مزیت اصلی Adam نسبت به الگوریتم‌های ساده‌تر مثل SGD این است که نرخ یادگیری را به صورت خودکار و مجزا برای هر پارامتر تنظیم می‌کند. پارامترهایی که تغییرات بیشتری دارند، نرخ بالاتری می‌گیرند. در نتیجه سرعت همگرایی و دقت مدل بهبود می‌یابد. اما SGD یک نرخ یادگیری ثابت برای همه پارامترهای مدل در نظر می‌گیرد،SGD به صورت تصادفی یک نمونه داده در هر مرحله انتخاب می‌کند در حالی که Adam از میانگین تمام گرادیان های مراحل قبل استفاده می‌کند.Adam به دلیل تنظیم خودکار نرخ یادگیری، کارایی و سرعت همگرایی بهتری دارد اما محاسبات بیشتری نسبت به SGD نیاز دارد.باتوجه به موارد ذکر شده در بالا بهینه ساز adam یک انتخاب مناسب برای پروژه میباشد.

تاثیر روش های Regularization:

Regularization با اعمال محدودیت بر پیچیدگی مدل ، از overfitting جلوگیری کرده و توانایی تعمیم‌پذیری مدل را بهبود می‌بخشد. این کار از طریق کاهش یا حذف وزن‌های کم‌اهمیت مدل انجام می‌شود تا مدل به جای یادگیری نویزهای موجود در داده‌های آموزشی ، الگوهای کلی‌تری را بیاموزد که بهتر قادر به تعمیم به داده‌های جدید است.

Drop Out:

به این صورت عمل میکند که در هر مرحله از اموزش مدل درصدی از نورون ها به صورت تصادفی غیر فعال میشوند که با عث میشود وابستگی مدل به برخی نورون های خاص کاهش یابد درنتیجه مدل قدرت تعمیم بیشتری دارد و overfitting کاهش میابد.

Batch Normalization:

با نرمالایزکردن خروجی هر لایه در شبکه عصبی، فرایند یادگیری را بهبود می‌بخشد. این الگوریتم با محاسبه میانگین و انحراف استاندارد داده‌های خروجی هر لایه در هر بچ، آن‌ها را به یک توزیع استاندارد نرمال می‌برد. این کار باعث می‌شود تا توزیع ورودی به هر لایه ، مستقل از تغییرات پارامترها در لایه‌های قبلی باشد و به وسیله re-scaling ، re-centering کردن ورودی ها به هر لایه از شبکه اجازه می دهد تا مستقل از لایه های دیگر آموزش ببینند. در نتیجه، هر لایه به طور مستقل و با سرعت بیشتری قادر به یادگیری خواهد بود. همچنین وابستگی به مقیاس اولیه پارامترها نیز برطرف می‌شود.

آموزش مدل و تاثیر پارامتر Batch Size:

Batch Size یکی از هایپرپارامترهای مهم در آموزش شبکه‌های عصبی عمیق است که تاثیر قابل توجهی بر کارایی، سرعت همگرایی و توانایی تعمیم‌دهی مدل دارد.

-مزایای اندازه‌ی بچ بزرگ:

الف) بهبود کارایی محاسباتی و بهره‌وری بهتر از سخت‌افزار GPU/TPU به دلیل پردازش موازی حجم بالاتری از داده

ب) کاهش واریانس گرادیان‌ها منجر به همگرایی پایدارتر می‌شود

- معایب اندازه‌ی بچ بزرگ:

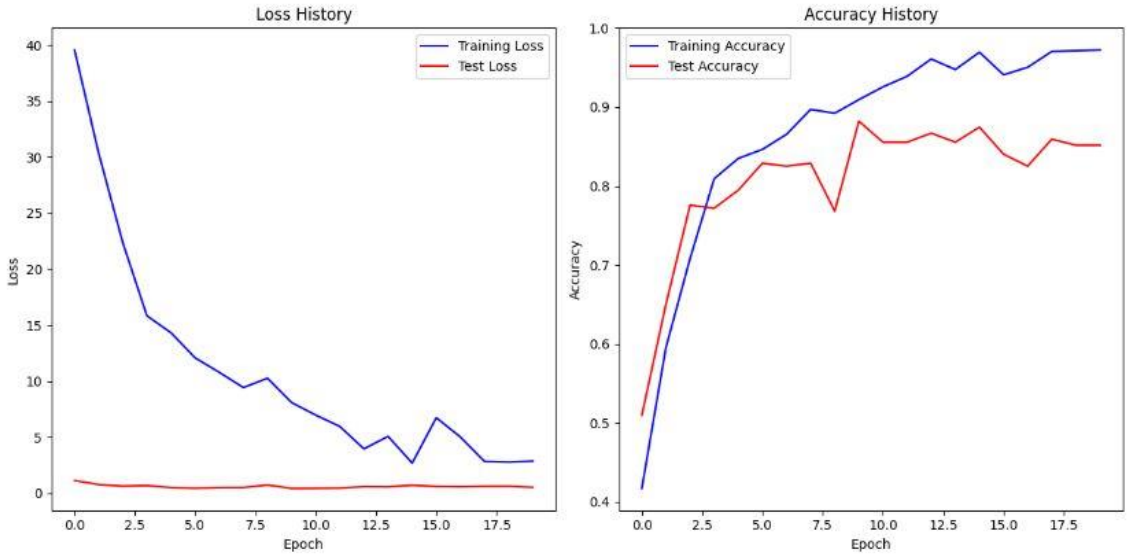
الف) منجر به کند شدن همگرایی و گیر افتادن در نقاط بهینه محلی شود

ب) توانایی تعمیم‌دهی به مجموعه آزمایش را کاهش دهد

پ) اندازه خیلی بزرگ می تواند باعث شود که مدل در یک بیشینه محلی گرفتار شود.

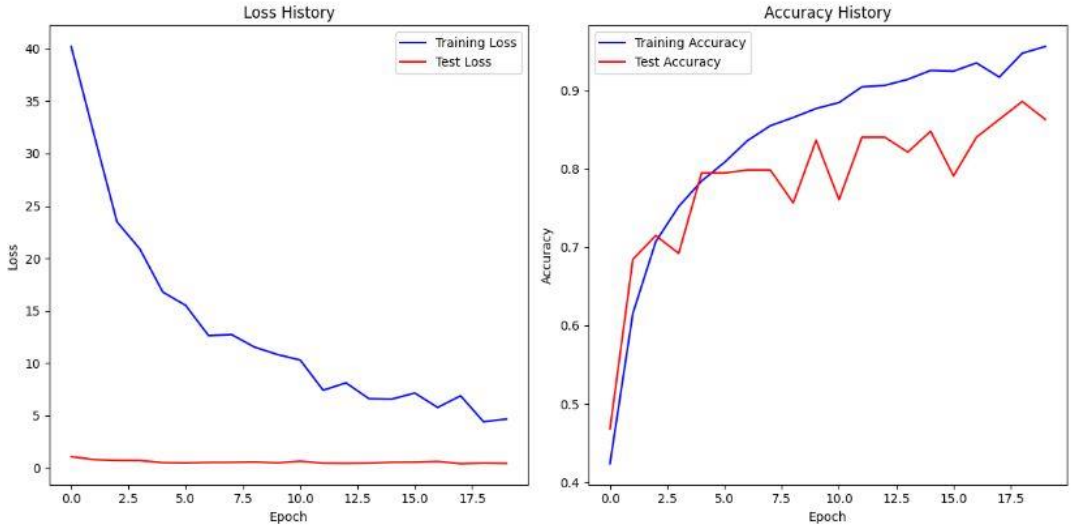
به همین علت در این پروژه میزان ۳۲ به عنوان اندازه بچ در نظر گرفته شد

نمودار بدون استفاده از Regularization :



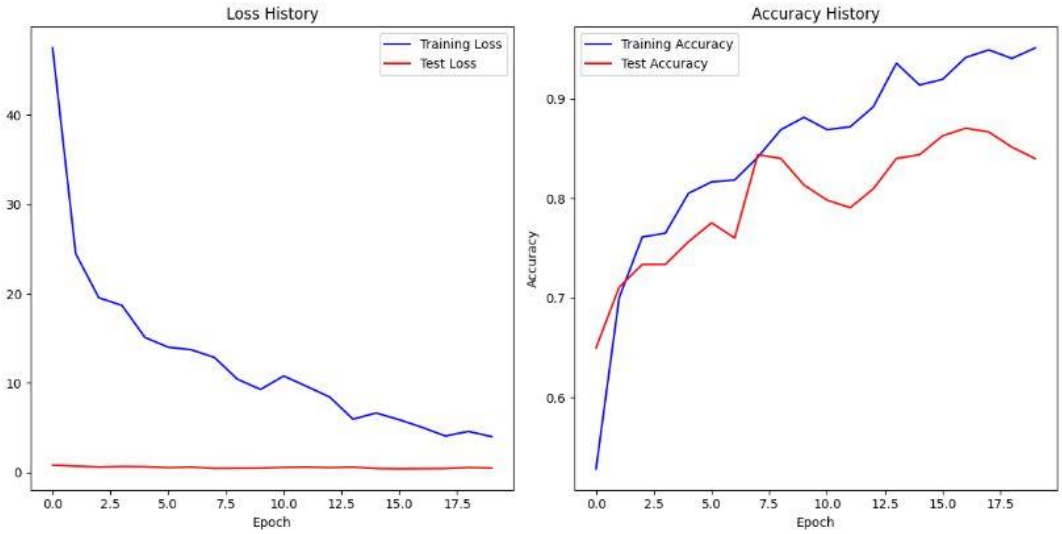
تاثیر dropout :

همانطور که انتظار میرفت سرعت همگرایی مدل کاهش یافت که باعث میشود مدل پایدار تر باشد و قدرت تعمیم بیشتر نیز داشته باشد مقادیر متفاوتی برای dropout بررسی شد درنهایت بهترین نتیجه با مقدار ۲۰ درصد به دست امد مقادیر بالاتر باعث میشدند که مدل به درستی اموزش نبیند و مقادیر پایین تر باعث میشدند که تاثیر مورد نیاز که از dropout انتظار داریم ایجاد نشود



تاثیر Batch Normalization :

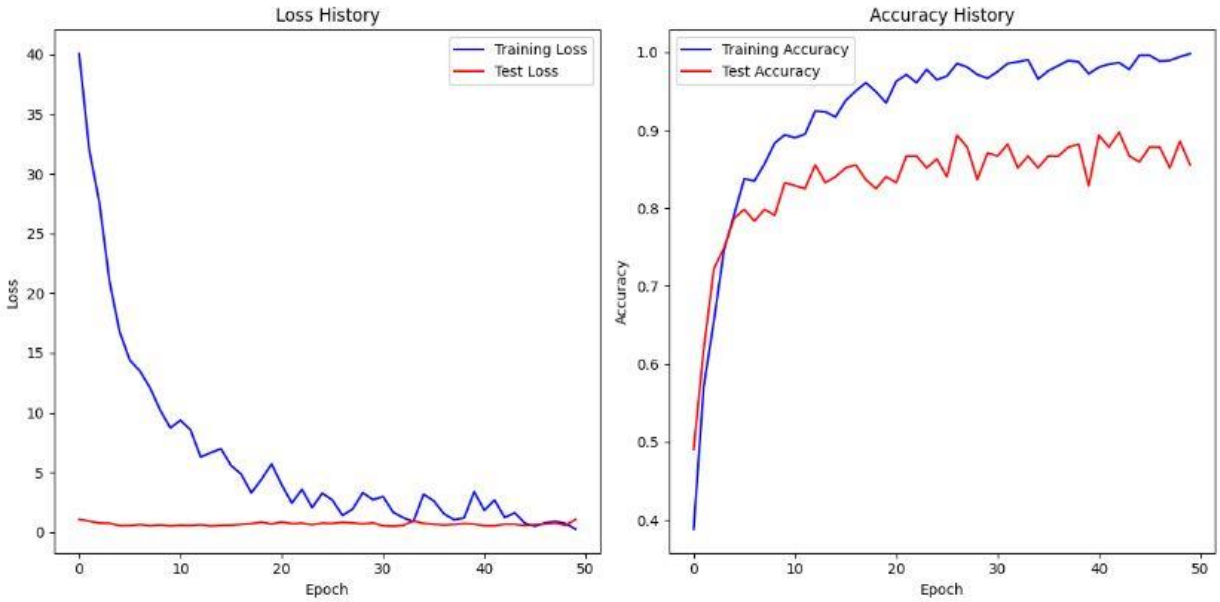
افزودن این لایه ها به معماری باعث شد که مدل به سرعت به یک مقدار همگرا شود و نتواند وضعیت خود را بهبود ببخشد در نتیجه مدل ها نسبت به حالت عادی دقت پایین تری خواهند داشت .در این بخش مشابه بخش قبل اندازه بچ های مختلف تست شد که دوباره بهترن نتیجه از اندازه ۳۲ حاصل شد

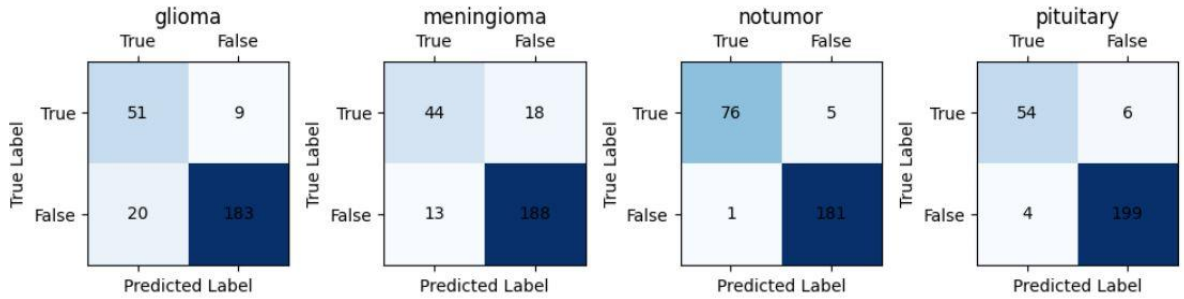


ارزیابی و تحلیل نتایج:

پس از بررسی و تغییر پارامتر ها بهترین معماری که به دست اوردیم معماری بالا به همراه ۲ لایه dropout با نرخ ۲۰ درصد در بخش تماما متصل میباشد که نتایج آن پس از ۵۰ epoch در زیر اماده است :

```
Best Model Accuracy: 89.73%
```





	glioma	meningioma	notumor	pituitary
Accuracy	51.695816	44.714828	76.688210	54.756653
Precision	0.718310	0.771930	0.987013	0.931035
Recall	0.850000	0.709677	0.938272	0.900000
F1 Score	0.778626	0.739496	0.962025	0.915254

```
Macro Avg {'Accuracy': 56.963876724243164, 'Precision': 0.8520717918872833, 'Recall': 0.8494872450828552, 'F1 Score': 0.8488503125477028}
```

همانطور که در عکس های بالا مشاهده میشود بهترین ممدل دارای دقت کلی ۸۹.۷۳ است همچنین این مدل عملکرد خوبی در تشخیص دارد اما در meningioma و glioma دقت پایین تری دارد همچنین loss پس از حدود ۴۵ epoch به صفر میل میکند و مدل با دقتی حدود ۸۵ همگرا میشود