دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

دانشکده مهندسی برق

گزارش تمرین سوم

Masked Autoencoder و Multi Modal

نگارش

محمدرضا شهرستانی ۴۰۳۱۱۵۰۸۳

حسین خموشی ۴۰۳۱۱۵۰۸۹

استاد

دکتر شریفیان

تیر ۱۴۰۴

چکیده

در این تمرین، مجموعه داده چندرسانه‌ای ROCOv2 شامل تصاویر رادیولوژی به همراه مفاهیم و عناوین پزشکی مرتبط مورد بررسی قرار گرفت. این مجموعه داده که از زیرمجموعه دسترسی آزاد PubMed استخراج شده است، شامل هفت روش بالینی مختلف بوده و مفاهیم آن به صورت دستی گردآوری و توسط متخصص رادیولوژی ارزیابی شده‌اند.

در مرحله بعد، مدل پیش‌آموزش دیده ViTMAE از کتابخانه Transformer بارگذاری شده و عملکرد آن در بازسازی تصاویر بخش Test با اعمال ۷۵ درصد ماسک مورد ارزیابی قرار گرفت. سپس این مدل با استفاده از داده‌های Train برای حوزه تصاویر پزشکی Fine Tune شد و بازسازی نمونه‌های تصاویر Test پس از آموزش مجدد بررسی گردید.

در بخش سوم، با استفاده از روش LoRA و کتابخانه PEFT، یک مدل چندرسانه‌ای برای تولید کپشن تصاویر پزشکی از مجموعه ROCOv2 ساخته و Fine Tune شد. عملکرد نهایی مدل با معیارهای استاندارد ارزیابی تصویر مانند CIDEr، BLEU، METEOR و ROUGE سنجیده شد. نتایج نشان‌دهنده توانایی این مدل‌ها در تحلیل و بازسازی تصاویر و تولید کپشن‌های مرتبط پزشکی بود.

لینک کدهای گزارش:

https://colab.research.google.com/drive/1kJd_EklY0U3rknEpLZq4Ly_JcZbXjsxB?usp=sharing

واژه‌های کلیدی:

Masked Autoencoder، Multi Modal، توضیح تصویر، LoRA

فهرست مطالب

صفحه

فهرست اشکال و جداول

صفحه

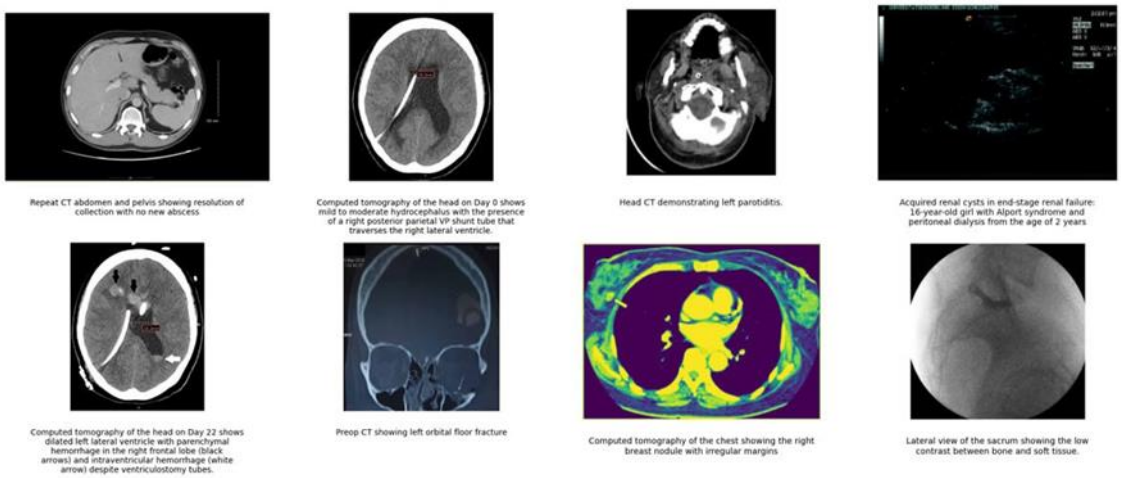
۱ مجموعه داده

۱-۱ بررسی مجموعه داده

مجموعه داده با نام Radiology Object in COntext (ROCOv2) در لینک زیر قرار داده شده است: [1]

https://zenodo.org/records/10821435

نمونه‌ای از تصاویر موجود در مجموعه‌داده را در تصویر ۱ ملاحظه می‌کنید:



تصویر ۱ – چند نمونه از تصاویر موجود در مجموعه داده

امروزه با توجه به پیشرفت‌هایی که در یادگیری عمیق رخ داده است، امکان تحلیل خودکار تصاویر پزشکی ممکن شده است. اما برای اینکار نیاز به داده‌های زیاد و باکیفیت داریم. در اینجا ما به بررسی مجموعه داده Radiology Object in COntext (ROCOv2) می‌پردازیم. دیتا ست ROCOv2 یک مجموعه داده‌ی چندوجهی شامل حدود ۸۰ هزار تصویر رادیولوژی و توضیحات و مفاهیم پزشکی مرتبط است که از مقالات دسترسی آزاد PubMed استخراج شده‌اند. از این تعداد ۳۵۷۰۵ تصویر در ورژن جدید این دیتاست اضافه شده است. در واقع این نسخه‌ی به‌روزشده‌ای از ROCO ۲۰۱۸ است که

شـــامل ۷ نوع تصـــویربرداری پزشـــکی بوده و برای آموزش مدل‌های بینایی را یانه‌ای در حوزه‌های برچســـب‌گذاری تصـــویر، تولید کپشـــن، یادگیری چندبرچســـبی، و یادگیری چندوظیفه‌ای کاربرد دارد. مفاهیم مرتبط با تصاویر (مانند آناتومی و جهت‌گیری) به‌صورت دستی توسط متخصصین تنظیم و ارزیابی شده‌اند. ساختار و محتویات آن بصورت دقیق‌تر شامل موارد زیر است:

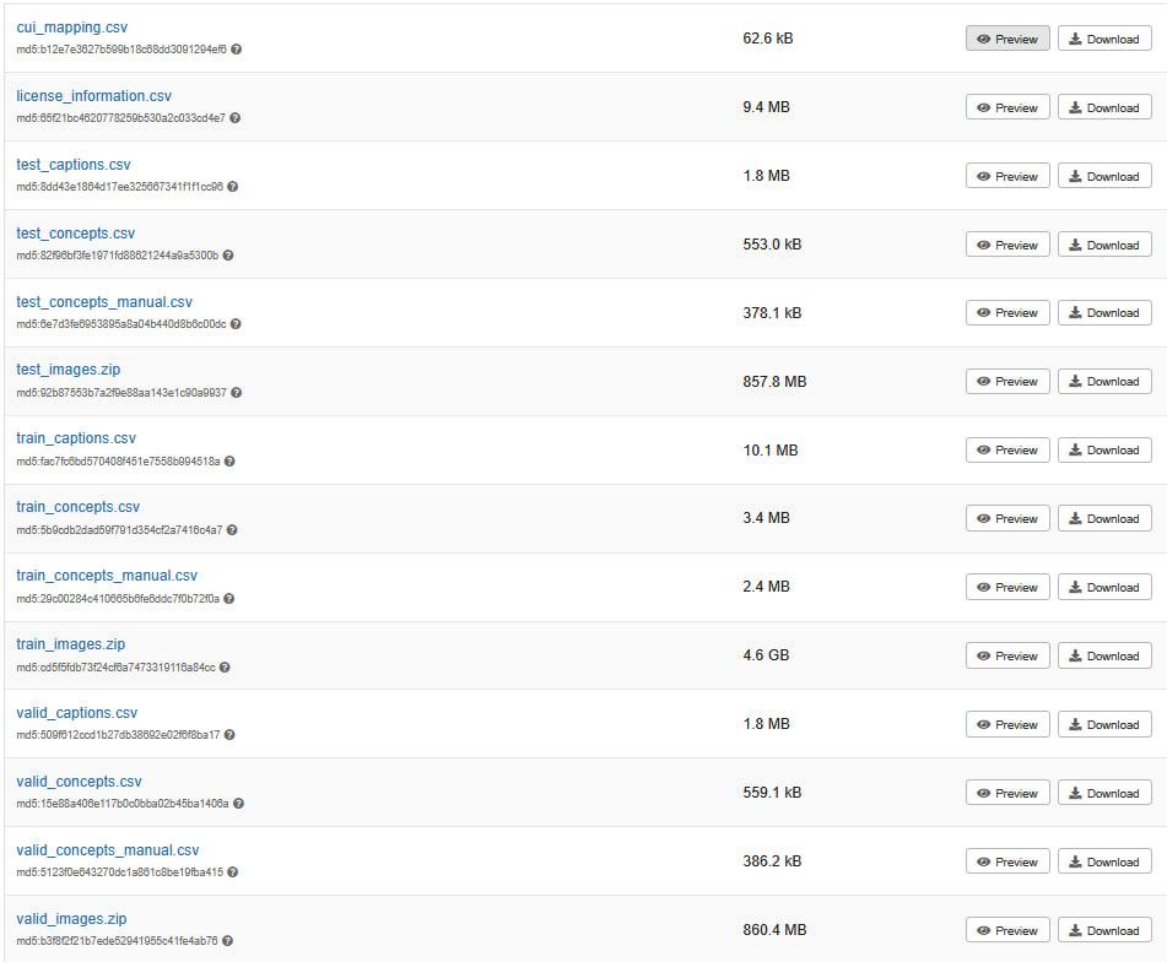
تصاویر: فریم‌برداری شده از داده‌های پزشکی واقعی (X-ray, CT, MRI)

متن‌ها: شرح‌های بالینی، شامل اصطلاحات استاندارد شده پزشکی (مثل CUI از UMLS).

جدول‌های کمکی: به‌عنوان مثال فایل cui_mapping.csv شـامل نگاشـت بین اصـطلاحات CUI و نام‌های رسمی آناتومیکی و بالینی است.

اگر بخواهیم در مورد نگارش و فرمت مجموعه داده صـــحبت می‌کنیم، می‌توانیم بگوییم که دیتاســـت به صـورت سـاخت‌یافته ارائه شـده و فایل‌هایی مانند cui_mapping.csv برای نگاشـت مفصـل مفاهیم استفاده شده است. فایل‌های متا شامل شناسه‌های استاندارد (CUI) و نام‌های رسمی آناتومی، بیماری‌ها و ضایعات هستند، که امکان تجزیه‌وتحلیل دقیق متنی را فراهم می‌کنند. هدف از ارائه این مجموعه داده، پیشبرد تحقیق در حوزه‌هایی مانند درک تصویر پزشکی، تولید شرح خودکار(captioning) ، جستجوی محتوا در میان تصاویر پزشکی، و آموزش مدل‌های هوش مصنوعی چندر سانه‌ای برای پردازش همزمان متن و تصویر است. علاوه بر این‌ها این مجموعه داده مزایایی دارد که بسیار به ما کمک می‌کند. از جمله آنها می‌توان مولتی‌مدال بودن، استاندارد بودن اصطلاحات را نام برد. مولتی‌مدال بودن برای ترکیب متن پزشکی با تصاویر رادیولوژی که برای آموزش مدل‌های پیشرفته مثل ViLT یا LXMERT استفاده می‌شود بسیار مناسب است. از طرفی استاندارد بودن اصطلاحات به این معنی است که استفاده از CUI و نگاشت دقیق اصطلاحات، برای تحلیل و یکپارچگی داده مفید است.

حالا به بررســـی دقیق‌تر فایل‌های که در مجموعه داده اســـت می‌پردازیم. در تصـــویر ۲ فایل‌هایی که در مجموعه داده قرار دارد را می‌بینید:



تصویر ۲ - فایل‌های موجود در مجموعه داده

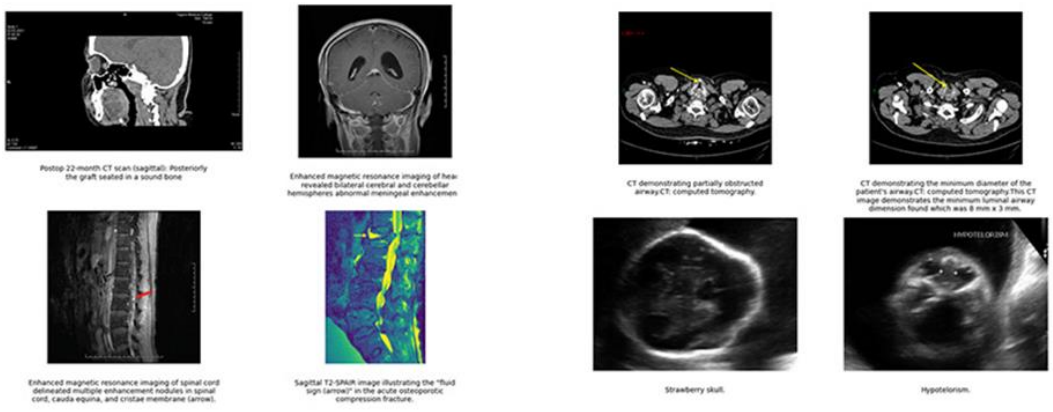
در این تمرین با توجه به مراحل بعدی تمرین، ما چهار فایل

- train_images.zip
- test_images.zip
- train_captions.csv
- test_captions.csv

را دانلود و load کرده‌ایم. در مراحل بعدی تمرین با این چهار فایل کار خواهیم کرد. برای مثال در قسمت ۲ تمرین، تصاویر موجود در فایل‌ها را load می‌کنیم و با یک مدل از پیش‌آموزش دیده به نام ViMAE از کتابخانه Transformer پنج نمونه از تصاویر مربوط به test را طبق شرایط تمرین باز سازی می‌کنیم. فایل zip با عنوان train_images.zip فایلی ا ست که ت صاویر مجموعه داده ما که

برای آموزش اســـتفـاده می‌کنیم در آن قرار دارد و حـدود ۴٫۶ گیگـابـایـت حجم دارد. فـایـل test_images.csv نیز تصـاویری قرار دارد که فقط برای test مدل اسـتفاده می‌شـود و حدود ۸۵۷٫۸ مگابایت حجم دارد. دو فایل با نام‌های train_captions.csv و test_captions.csv داریم که این دو mapping بین تصاویر و caption تصاویر هستند. در واقع توضیحات مربوط به هر تصویر داخل این فایل‌های csv هســـتند. همانطور که از نام فایل‌ها پیداســـت، caption مربوط به تصـــاویری که در train_images.zip قرار دارد در فایل train_captions.csv است. caption مربوط به تصاویر test نیز در فایل test_images.csv قرار دارد. بطور خلا صه ما دو فایل zip داریم که تصاویر برای train و test در آن قرار دارد و دو فایل csv متناظر این‌ها را داریم که شرح تصاویر مجموعه train و test است.

در مجموع، ROCOv2 که یک بروزرسانی برای ورژن ۱ آن است، یک منبع ارزشمند برای پژوهشگران حوزه هوش مصنوعی در پزشکی است که نیازمند داده‌های ترکیبی تصویر و متن هستند؛ به ویژه برای کاربردهایی مانند تشخیص خودکار، شرح، تفکیک ساختاری یا تحلیل زمینه‌ای در تصاویر پزشکی. چند نمونه دیگر از تصاویر مجموعه داده را در تصویر ۳ مشاده می‌کنید:



تصویر ۳ – چند نمونه دیگر از تصاویر موجود در مجموعه داده

Masked Autoencoder ۲

۲-۱ مدل از پیش آموزش دیده

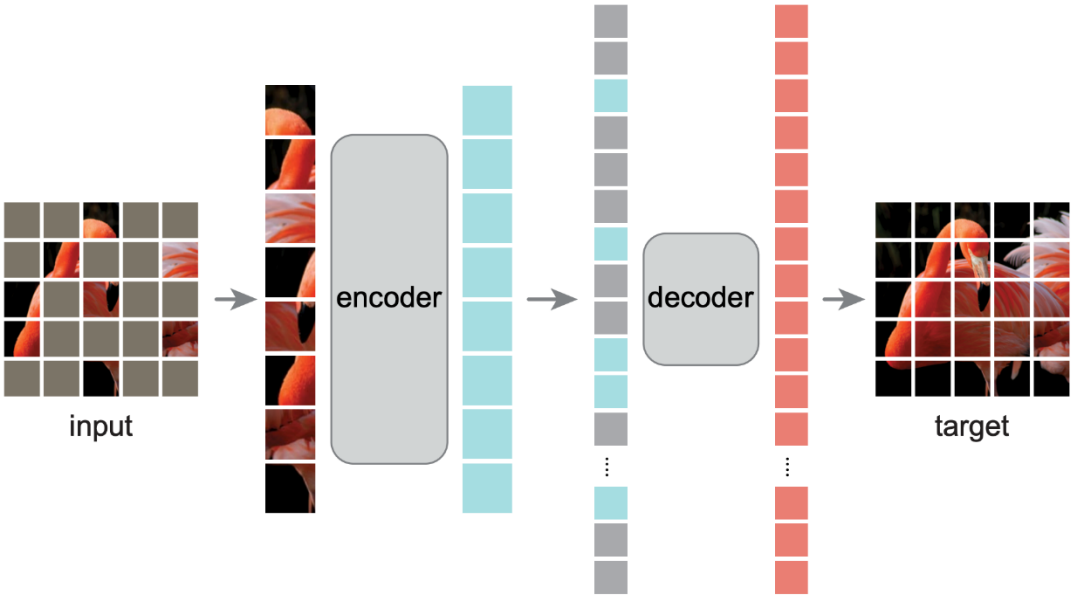
طبق خواسته تمرین، ابتدا مجموعه‌داده‌های آموزش و آزمون را بارگیری و نمایش می‌دهیم. تصویر ۱ و ۳ و ۴ از مجموعه داده‌ی آموزشی با نگاشت تصویر و توضیح آن است که در این بخش به توضیحات نیاز نداریم.

```
test  test_captions.csv  train  train_captions.csv
--------------------------------------------------
ROCOv2_2023_train_000001.jpg
ROCOv2_2023_train_000002.jpg
ROCOv2_2023_train_000003.jpg
ROCOv2_2023_train_000004.jpg
ROCOv2_2023_train_000005.jpg
ROCOv2_2023_train_000006.jpg
ROCOv2_2023_train_000007.jpg
ROCOv2_2023_train_000008.jpg
ROCOv2_2023_train_000009.jpg
ROCOv2_2023_train_000010.jpg
--------------------------------------------------
ROCOv2_2023_test_000001.jpg
ROCOv2_2023_test_000002.jpg
ROCOv2_2023_test_000003.jpg
ROCOv2_2023_test_000004.jpg
ROCOv2_2023_test_000005.jpg
ROCOv2_2023_test_000006.jpg
ROCOv2_2023_test_000007.jpg
ROCOv2_2023_test_000008.jpg
ROCOv2_2023_test_000009.jpg
ROCOv2_2023_test_000010.jpg
--------------------------------------------------
                       ID                                            Caption
0  ROCOv2_2023_train_000001              Head CT demonstrating left parotiditis.
1  ROCOv2_2023_train_000002  Acquired renal cysts in end-stage renal failur...
2  ROCOv2_2023_train_000003  Computed tomography of the chest showing the r...
3  ROCOv2_2023_train_000004  Lateral view of the sacrum showing the low con...
4  ROCOv2_2023_train_000005  Thoracic CT scan showing perihilar pulmonary l...
5  ROCOv2_2023_train_000006  5.1 cm x 3.4 cm x 4 cm multiloculated hepatic ...
6  ROCOv2_2023_train_000007  Repeat CT abdomen and pelvis showing resolutio...
7  ROCOv2_2023_train_000008  Computed tomography of the head on Day 0 shows...
8  ROCOv2_2023_train_000009  Computed tomography of the head on Day 22 show...
9  ROCOv2_2023_train_000010       Preop CT showing left orbital floor fracture
--------------------------------------------------
                       ID                                            Caption
0  ROCOv2_2023_test_000001  CT chest axial view showing a huge ascending a...
1  ROCOv2_2023_test_000002  Computed tomography (CT) shows floating thromb...
2  ROCOv2_2023_test_000003  Digitally subtracted angiogram demonstrates ac...
3  ROCOv2_2023_test_000004  Digitally subtracted angiogram of the IMA demo...
4  ROCOv2_2023_test_000005               Angle measurement of a Type 1 canal.
5  ROCOv2_2023_test_000006  Computed tomography on day 26Follow-up enhance...
6  ROCOv2_2023_test_000007  Enhanced CT scan of the chest revealed an ante...
7  ROCOv2_2023_test_000008                Arrow shows ULP at the distal arch.
8  ROCOv2_2023_test_000009                    Early sagittal T2-weighted MRI.
9  ROCOv2_2023_test_000010                       Late axial T2-weighted MRI.
```

تصویر ۴ – بارگیری مجموعه‌ی داده تصلویر همراه با توضیحات در محیط اجرا

مدل از پیش آموزش دیده‌ی ViTMAE از کتابخانه Transformer را load کردیم. ۵ تصویر اول را بارگذاری کردیم. با استفاده از ViTFeatureExtractor تصویر را به شکلی تبدیل می‌کنیم که مدل بتواند آن را پردازش کند. برای مثال برش‌های مربوط به patchها، نرمال‌سازی، تغییر سایز و... را دیگر می‌توانیم انجام دهیم. [2]

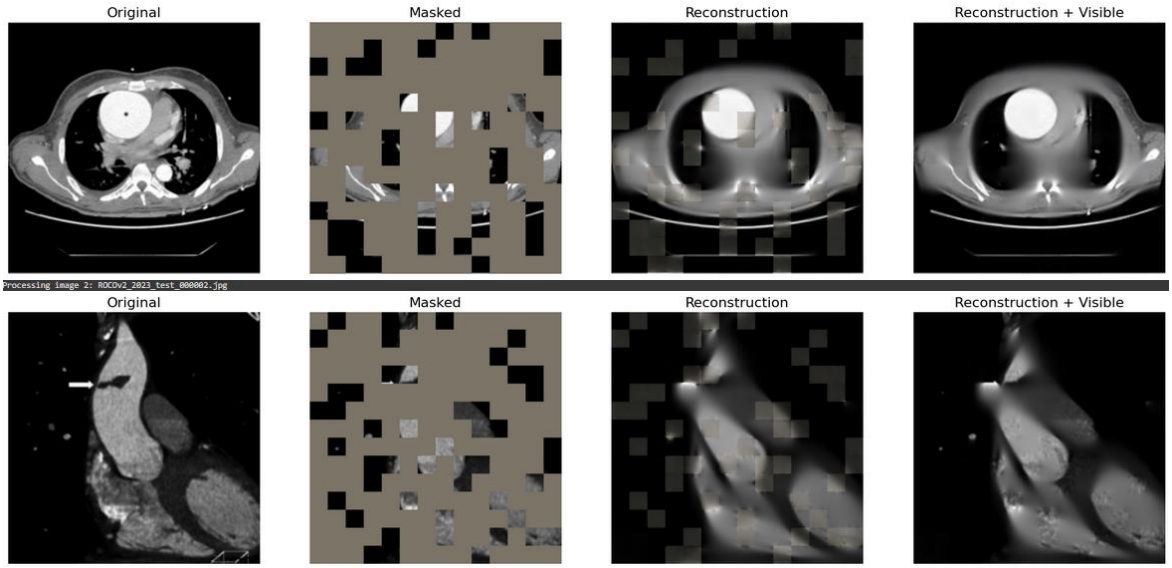
تصویر ۵ شمایی از معماری مدل را نشان می‌دهد.



تصویر ۵ – معماری ویژن ترنسفورمری مدل ViTMAE با ورودی قطعات patch

همچنین ما درصد ماسک کردن را طبق گفته تمرین ۷۵٪ در نظر گرفتیم؛ یعنی ۷۵٪ تصویر ماسک می‌شود و فقط ۲۵٪ تصویر پیداست. سپس ما به سراغ تابع visualize می‌رویم. در این تابع فرایند اصلی اجرا می‌شود. در این تابع ابتدا تصویر به مدل داده می‌شود؛ سپس patchهای بازسازی‌شده به تصویر بازسازی‌شده تبدیل می‌شوند. ماسک استفاده‌شده نیز بازسازی می‌شود. سپس روی پنج تصویر از پوشه

test اعمال می‌کنیم. در مرحله بعدی چهار تصویر مختلف رسم می‌شود: Orginal، Masked، Reconstruction و Reconstruction + Visible. چند نمونه از آن را می‌توانید در تصویر ۶ ببینید.



تصویر ۶ - تصویر اصلی، ماسک شده، همراه با تصاویر بازسازی شده

با توجه به خروجی که در تصویر ۶ می‌بینید از لحاظ بصری بنظر با کیفیت قابل قبولی تصاویر باز سازی شده‌اند. البته اگر دقیقتر نگاه کنیم در مثال اول کیفیت تصویر بازسازی‌شده بهتر است و به نسخه original آن نزدیک‌تر است.

در ادامه سعی می‌کنیم با نوشتن کدی، کیفیت تصاویر باز سازی شده را با معیارهای کمّی بسنجیم. پس در این مرحله ابتدا تصویر را به مدل ViT-MAE می‌دهیم. سپس تصویر باز سازی شده را نیز بدست می‌آوریم. حالا با استفاده از سه معیار زیر کیفیت تصاویر بازسازی‌شده را نسبت به تصویر اصلی اندازه‌گیری می‌کنیم:

- MSE (میانگین مربع خطا)
- PSNR (نسبت سیگنال به نویز پیک)
- SSIM (شاخص شباهت ساختاری)

به توضیح هر یک از این شاخص‌ها می‌پردازیم.

MSE: اختلاف بین پیکسل‌های تصویر اصلی و تصویر باز سازی شده را به صورت مربع میانگین‌گیری می‌کند. هر چه به صفر نزدیک‌تر باشد بهتر است و به معنی این است که تصویر بازسازی‌شده شبیه تصویر اصلی است. اگر دو تصویر دقیقا یکسان باشند، MSE برابر صفر خواهد بود.

PSNR: نسبت ماکزیمم توان سیگنال (پیکسل) به توان خطا (نویز) است. PSNR بالا یعنی نویز کمتر و شباهت بیشتر. واحد آن بر اساس dB است. اگر دو تصویر دقیقا یکسان باشند، PSNR بی‌نهایت است.

SSIM: یک معیار پیچیده‌تر است که شباهت ساختاری، روشنایی و کنتراست بین دو تصویر را بررسی می‌کند. در واقع شاخصی ساختاری که شباهت بین دو تصویر را در سطح درک انسان اندازه‌گیری می‌کند، است. مقدار نزدیک به ۱ بهتر است. در کل این معیار بین ۰ تا ۱ است.

حالا به ادامه تحلیل می‌پردازیم. طبق کدی که نوشتیم و خروجی که گرفتیم معیارهای زیر را بررسی می‌کنیم:

```
Processing image 1: ROCOv2_2023_test_000001.jpg
Processing image 2: ROCOv2_2023_test_000002.jpg
Processing image 3: ROCOv2_2023_test_000003.jpg
Processing image 4: ROCOv2_2023_test_000004.jpg
Processing image 5: ROCOv2_2023_test_000005.jpg

--- Quantitative Evaluation Results ---
Image                          MSE        PSNR       SSIM
ROCOv2_2023_test_000001.jpg    2.6912     -4.30      0.0263
ROCOv2_2023_test_000002.jpg    2.6820     -4.28      0.0080
ROCOv2_2023_test_000003.jpg    0.1198     9.21       0.2557
ROCOv2_2023_test_000004.jpg    0.0296     15.29      0.3301
ROCOv2_2023_test_000005.jpg    0.9970     0.01       0.0636
```

تصویر ۷ - خروجی برای مقایسه معیارها

طبق نتایج بدست آمده در تصویر ۷، به بررسی می‌پردازیم. ابتدا سراغ معیار MSE می‌رویم. معیار MSE همانطور که پیش‌تر توضیح دادیم هر چه به صفر نزدیک‌تر باشد بهتر است. طبق نتایج می‌بینید که MSE دو داده‌ی اول حدود ۲,۶ است و تصاویر به خوبی بازسازی نشده‌اند. اما داده‌های ۳، ۴ و ۵ نسبت به دو داده‌ی اولی با کیفیت خوبی بازسازی شده‌اند. حالا به سراغ معیار PSNR می‌رویم. دو تصویر اول این معیار منفی شده و حدود ۴- است. که نشان می‌دهد. دقت کنید که PSNR هر چقدر بالاتر باشد بهتر است. همانطور که در نتایج می‌بینید دو تصویر اول PSNR پایینی دارند تصویر سوم و

چهارم به مراتب بهتر هستند و تصویر پنجم نزدیک به صفر است که این هم کیفیت مناسبی نیست. اما به سراغ شاخص سوم که SSIM باشد می‌رویم. همانطور که گفتیم SSIM به ما کمک می‌کند که شباهت دو تصویر را در سطح درک انسان اندازه‌گیری کنیم. در این شاخص همانطور که در نتایج پیداست دو تصویر سوم و چهارم عملکرد نسبتا بهتری نسبت به بقیه تصاویر داشته‌اند. ضعیف‌ترین نتیجه در این معیار نیز مربوط به تصویر دوم بوده است که به این معناست، تصویر بازسازی‌شده از لحاظ شباهت ساختاری، نور، کنتراست و... نسبت به تصویر اصلی شباهت کمی دارد.

Fine Tune ۲-۲

در این قسمت آمدیم مدلی که ذخیره کرده بودیم را reload و fine-tune کردیم. در ابتدا بعد از بارگذاری مدل، نرخ mask را روی ۷۵٪ تنظیم کردیم. سپس یک دیتاست کاستوم درست کردیم. ابتدا ۴۰۰۰ تصویر اول را از مسیر train/ بارگذاری می‌کنیم. ما هر تصویر را به RGB تبدیل می‌کنیم و سپس با feature_extractor پیش‌پردازش‌های لازم (normalize، resize، تبدیل به tensor) را انجام می‌دهیم. خروجی‌هایی که تولید می‌شود، ورودی مناسب برای مدل ViT خواهد بود. سپس این دیتاست کاستوم را بارگذاری می‌کنیم و تنظیمات training را انجام می‌دهیم. برای train، یکسری پارامتر مهم تنظیم می‌کنیم که تعدادی از آن‌ها به شرح زیر است:

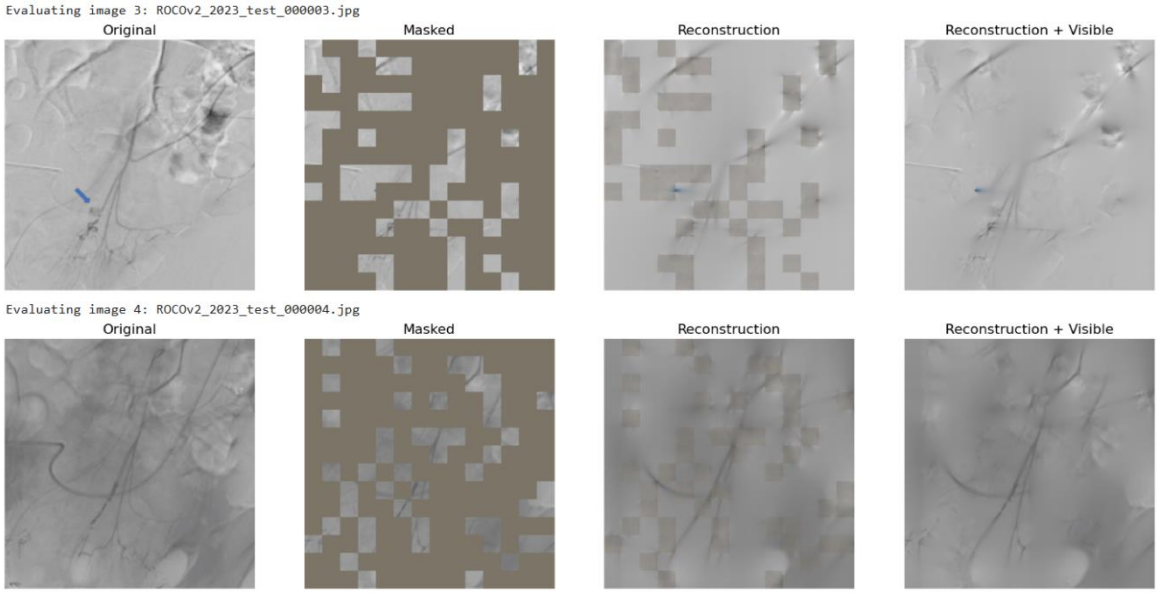
- اندازه هر batch را تعیین می‌کنیم که اینجا هر batch شامل ۸ تصویر است.
- تعداد epochها را تنظیم می‌کنیم که اینجا ما ۵ epoch (حداقل تعدادی که در صورت تمرین ذکر شده بود) را در نظر گرفتیم.
- بعد از هر epoch نیز ذخیره می‌کنیم.
- پارامتر save_total_limit را مساوی ۱ قرار دادیم که فقط آخرین مدل ذخیره شود.
- پارامتر remove_unused_columns را False قرار دادیم برای حفظ compatibility.
- پارامتر report_to نیز none تنظیم شده است.[۱]

۱ از گزارش‌دهی به WandB یا سایر سرویس‌ها جلوگیری می‌کند. WandB یک ابزار مانیتورینگ و لاگ‌گیری پروژه‌های یادگیری ماشین است.

حالا Trainer را که از کتابخانه HuggingFace است برای آموزش مدل استفاده می‌کنیم و پارامترهای لازم را به عنوان ورودی به آن می‌دهیم. در نهایت نیز مدل آموزش‌دیده را ذخیره می‌کنیم.

در مجموع در این قسمت ما یک مدل ViT-MAE را که از قبل وجود داشته است با داده‌های خود آموزش مجدد دادیم و نتیجه را ذخیره کردیم.

بعد از این مرحله سراغ ارزیابی کیفی (بصری) مدل ViT-MAE روی ۵ داده‌ی اول از پوشه test/ می‌رویم. در واقع مدل تلاش می‌کند قسمت‌های پنهان شده (ماسک شده) تصاویر را بازسازی کند و نتیجه را به همراه تصویر اصلی نشان دهد. در واقع چهار خروجی Original، Masked، Reconstruction و Reconstruction + Visible نمایش داده می‌شود. ابتدا کارهای لازم از جمله تبدیل به RGB و... را انجام می‌دهد سپس تصویر را به مدل می‌دهد تا بازسازی کند و همانطور که ذکر شد چهار خروجی به ما می‌دهد. در تصویر ۸، دو نمونه را می‌توانید ببینید.



تصویر ۸ - خروجی روی داده‌های test

همانطور که در خروجی می‌بینید در حالت بصری، تصاویر با کیفیت نسبتا قابل قبولی بازسازی شده‌اند.

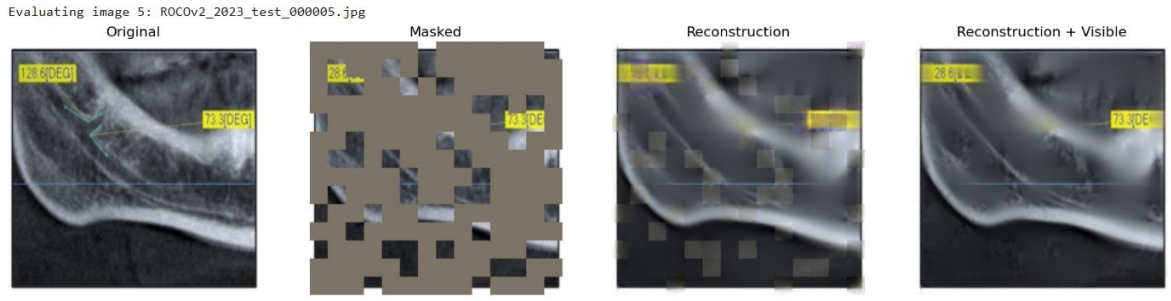
حالا دوباره مانند قسمت قبل تمرین، به ارزیابی تصاویر با سه معیار MSE، PSNR و SSIM می‌پردازیم. خروجی ارزیابی در تصویر ۹ آمده است.

```
Processing image 1: ROCOv2_2023_test_000001.jpg
Processing image 2: ROCOv2_2023_test_000002.jpg
Processing image 3: ROCOv2_2023_test_000003.jpg
Processing image 4: ROCOv2_2023_test_000004.jpg
Processing image 5: ROCOv2_2023_test_000005.jpg

--- Quantitative Evaluation Results ---
Image                          MSE        PSNR       SSIM
ROCOv2_2023_test_000001.jpg    2.6620     -4.25      0.0313
ROCOv2_2023_test_000002.jpg    2.6835     -4.29      0.0077
ROCOv2_2023_test_000003.jpg    0.1210     9.17       0.2506
ROCOv2_2023_test_000004.jpg    0.0328     14.84      0.3148
ROCOv2_2023_test_000005.jpg    0.9946     0.02       0.0663
```

تصویر ۹ - نتایج کمی ارزیابی رو داده‌های test

همانطور که از نتایج پیداست در معیار MSE دو تصویر بازسازی‌شده اول و دوم، نتایج ضعیفتری داشته‌اند. بهترین نتیجه نیز مربوط به تصویر چهارم است. در معیار PSNR نیز دو تصویر اول و دوم منفی شده‌اند و نتیجه ضعیفی است اما تصاویر بازسازی‌شده سوم و چهارم، نتیجه بهتری داشته‌اند. تصویر پنجم نیز عددی نزدیک به صفر است که نتیجه نسبتا ضعیفی است. در معیار SSIM، تصاویر سوم و چهارم عملکرد نسبتا خوبی داشته‌اند در حالی که تصاویر اول، دوم و پنجم عملکرد تقریبا ضعیفی داشته‌اند. ضعیفترین عملکرد مربوط به تصویر دوم در این معیار بوده است. یک مثال دیگر از خروجی بصری را در تصویر ۱۰ می‌بینید.



تصویر ۱۰ - خروجی بصری روی داده‌های test

همانطور که در تصویر می‌بینید از لحاظ بصری، خروجی قابل قبولی تولید شده است.

Multi Modal ۳

۳-۱ مدل از پیش آموزش دیده

ما مدل ViT-GPT2 با وزن های Pretrain شـــده‌ی google/vit-base-patch16-224-in21k را انتخاب کردیم.[3]

مدل vit-base-patch16-224-in21k یکی از نسخه‌های پایه معماری Vision Transformer (ViT) است که توسط شرکت گوگل ارائه شده و بر روی مجموعه‌داده‌ی بسیار بزرگ ImageNet-21k آموزش دیده است. این مدل از معماری ترنسفورمر، که پیش‌تر در حوزه‌ی پردازش زبان طبیعی بسیار موفق عمل کرده بود، برای تحلیل تصاویر استفاده می‌کند. برخلاف شبکه‌های عصبی پیچشی که به‌صورت موضعی و لایه‌به‌لایه ویژگی‌های تصویر را استخراج می‌کنند، مدل ViT تصویر را به قطعات (پچ‌ها) تقسیم می‌کند و آن‌ها را مانند توکن‌های متنی در زبان طبیعی به مدل ترنسفورمر می‌دهد.

در نسخه‌ی vit-base-patch16-224-in21k، تصویر ورودی دارای اندازه‌ی ۲۲۴×۲۲۴ پیکسل است و به پچ‌هایی با اندازه‌ی ۱۶×۱۶ تقسیم می‌شود. در نتیجه، تصویر به ۱۴×۱۴=۱۹۶ پچ تقسیم می‌شود. هر پچ با یک لایه‌ی خطی یک بردار عددی با بعد ثابت تبدیل می‌شـــود و ســـپس به مدل داده می‌شـــود. همچنین یک توکن ویژه به نام CLS به ابتدای این دنباله اضافه می‌شود که نماینده‌ی کل تصویر است و خروجی آن برای انجام وظیفه‌ی نهایی مانند دسته‌بندی یا استخراج ویژگی استفاده می‌شود.

برای آن‌که مدل بتواند موقعیت نســـبی پچ‌ها را تشـــخیص دهد، به هر پچ embedding موقعیت (position embedding) اضافه می‌شود. سپس این دنباله‌ی توکن‌ها وارد بلوک‌های ترنسفورمر می‌شود که شامل attention چندسَری، لایه‌های normalization، و شبکه‌های عصبی feed-forward هستند. مدل base از ۱۲ بلوک ترنسفورمر تشکیل شده و در هر بلوک ۱۲ سر attention وجود دارد، که همگی به‌صورت کاملاً مشابه ترنسفورمر در زبان عمل می‌کنند، اما بر روی پچ‌های تصویری.

مدل vit-base-patch16-224-in21k به‌صورت ویژه برای استخراج ویژگی از تصویر طراحی شده است، نه برای وظیفه‌ی خاصـــی مانند دســـته‌بندی نهایی. این به آن معناســـت که این مدل بیشـــتر در نقش encoder استفاده می‌شود و می‌توان آن را در ترکیب با یک decoder زبانی (مانند GPT2 یا T5) به‌کار

گرفت تا وظیفه‌هایی مانند تولید توضیح متنی از تصویر انجام شود. برای همین در کاربردهایی که به اتصال بین تصویر و متن نیاز دارند، این مدل به‌عنوان پایه‌ی پردازش تصویر استفاده می‌شود، چون ویژگی‌های غنی و عمومی تولید می‌کند.

در مجموع، vit-base-patch16-224-in21k یک مدل قدرتمند و انعطاف‌پذیر برای پردازش بینایی ماشین است که به‌دلیل آموزش روی مجموعه‌داده‌ی بسیار بزرگ، در بسیاری از وظایف پایین‌دستی مانند تشخیص تصویر، توصیف تصویر، شناسایی اشیا، و حتی رباتیک می‌تواند به‌عنوان encoder پایه مورد استفاده قرار گیرد. همچنین با استفاده از روش‌های بهینه‌سازی سبک مانند LoRA، می‌توان این مدل را با منابع محاسباتی کم نیز برای وظایف خاص خود تطبیق داد.

هدف ما ساخت یک مدل تبدیل تصویر به متن برای توضیح تصاویر است که از دو مدل پیش‌آموزش‌دیده استفاده می‌کند: یک مدل Vision Transformer برای استخراج ویژگی‌های تصویر و یک مدل زبانی DistilGPT2 برای تولید متن. این ساختار ترکیبی به‌صورت encoder–decoder طراحی شده است و برای آموزش بهتر و کاراتر، آماده‌ی استفاده با روش LoRA نیز هست.

در ابتدا، کتابخانه‌ها و ابزارهای موردنیاز وارد می‌شوند. این‌ها شامل torch برای پردازش PyTorch، PIL برای خواندن تصاویر، tqdm برای پیشرفت نمایی، و transformers برای استفاده از مدل‌های HuggingFace هستند. همچنین با استفاده از warnings.filterwarnings("ignore") هشدارها خاموش می‌شوند تا اجرای کد خللی نداشته باشد.

سپس دو مدل اصلی مشخص می‌شوند:

encoder_model_id = "google/vit-base-patch16-224-in21k" که وظیفه‌ی تحلیل تصویر را بر عهده دارد و از نوع ViT (Vision Transformer) است.

decoder_model_id = "distilgpt2" که وظیفه‌ی تولید متن را بر اساس ویژگی‌های استخراج‌شده از تصویر دارد و نسخه‌ی سبک‌شده‌ای از GPT2 است.

در گام بعدی، از AutoFeatureExtractor برای آماده‌سازی تصاویر استفاده می‌شود. این ابزار تصویر را به فرمت مناسب برای مدل ViT تبدیل می‌کند (مانند تغییر اندازه، نرمال سازی و تبدیل به تنسور). در

همین حال، از AutoTokenizer برای آماده‌سازی متن استفاده می‌شود و اگر توکن پرکننده (pad token) برای GPT2 تعریف نشده باشد، به صورت دستی توکن پایان (EOS) به‌عنوان توکن پرکننده تعیین می‌شود؛ این کار برای تضمین عملکرد درست در مراحل آموزش و تولید متن ضروری است.

در مرحله‌ی بعد، از کلاس VisionEncoderDecoderModel استفاده می‌شود تا encoder و decoder را به هم متصل کنیم و یک مدل کامل Image Captioning بسازیم. این مدل از ViT برای استخراج ویژگی و از DistilGPT2 برای تولید جمله استفاده می‌کند. استفاده از torch_dtype=torch.float16 نیز باعث می شود که حافظه‌ی کمتری مصرف شود و سرعت پردازش افزایش یابد.

در پایان، پارامترهای لازم برای تولید متن پیکربندی می‌شوند. decoder_start_token_id مشخص می‌کند تولید متن از چه توکنی آغاز شود (معمولاً CLS یا .BOS) سپس pad_token_id برای پر کردن خروجی‌های کوتاه‌تر تنظیم می شود. در نهایت vocab_size برای تطبیق ابعاد خروجی decoder با اندازه‌ی واژگان آن مشخص می‌گردد. این تنظیمات، مدل را برای انجام وظایف captioning یا سایر وظایف تولید متن از تصویر آماده می‌کنند.

۳-۲ Fine Tune بهینه‌ی پارامترها با LoRA

یکی از دلایل اصلی استفاده از تکنیک LoRA، کاهش حجم مدل است. در این روش، تنها بخشی از پارامترهای مدل برای آموزش تنظیم می‌شوند که این موضوع به‌طور قابل توجهی حجم کلی مدل را کاهش می‌دهد.همچنین، استفاده از LoRA موجب افزایش بهره‌وری می‌شود. با کاهش تعداد پارامترهای قابل آموزش، زمان و منابع محاسباتی مورد نیاز برای آموزش کاهش یافته و در نتیجه فرآیند آموزش سریع‌تر و کارآمدتر انجام می‌شود.

به طور کلی، استفاده از تکنیک LoRA این امکان را فراهم می‌آورد که مدل‌های بزرگ و پیچیده را با منابع کمتر و بهره‌وری بیشتر آموزش دهیم و در نهایت عملکرد مدل را بهبود ببخشیم. (تصویر ۱۱)

Fine-tune Model with LoRA

```python
import warnings
warnings.filterwarnings("ignore")

# Configure LoRA for parameter-efficient fine-tuning
lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["c_attn", "c_proj"], # Apply LoRA to attention projections in the decoder (GPT-2)
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM"
)

# Apply LoRA to the base model
model = get_peft_model(model, lora_config)

# Print trainable parameters after LoRA
print("Trainable parameters after applying LoRA:")
model.print_trainable_parameters()
```

```
Trainable parameters after applying LoRA:
trainable params: 1,179,648 || all params: 183,664,896 || trainable%: 0.6423
```

تصویر ۱۱ – استفاده از تکنیک LoRA برای بهینه‌سازی پارامترها

در ابتدا با استفاده از کلاس LoraConfig تنظیمات LoRA مشخص می‌شود. مقدار r=16 و lora_alpha=32 بیانگر رتبه پایین و scale factor برای ماتریس‌های LoRA هستند. آرگومان target_modules=["c_attn", "c_proj"] تعیین می‌کند که LoRA فقط روی لایه‌های projection مربوط به attention در مدل GPT-2 اعمال شود؛ یعنی روی لایه‌های ترکیب و خروجی attention در decoder (که مربوط به تولید متن هستند). با lora_dropout=0.05 نیز نرخ dropout مشخص شده تا در حین آموزش از overfitting جلوگیری شود. گزینه‌ی task_type="CAUSAL_LM" هم نشان می‌دهد که مدل برای وظیفه‌ی مدل‌سازی زبانی علّی (مانند GPT2) تنظیم می‌شود.

سپس با تابع get_peft_model، این پیکربندی LoRA روی مدل پایه (VisionEncoderDecoder) اعمال می‌شود. این مرحله باعث می‌شود که به مدل لایه‌هایی با پارامترهای اضافه‌شده‌ی کم‌حجم افزوده شود که فقط همان‌ها در طول آموزش تغییر می‌کنند، در حالی که باقی پارامترهای مدل ثابت باقی می‌مانند.

در پایان، با اجرای model.print_trainable_parameters() تعداد دقیق پارامترهایی که قابل آموزش هستند چاپ می‌شود.

پس از اعمال روش LoRA روی مدل ترکیبیViT-GPT2 ، تنها حدود **۱,۱ میلیون پارامتر** از مجموع **۱۸۳ میلیون پارامتر** مدل به حالت آموزش‌پذیر درآمده‌اند که معادل تنها **٪۰,۶۴** از کل پارامترهاست. این نشان می‌دهد که با استفاده ازLoRA ، می‌توان مدل را به‌صورت بسیار کم‌هزینه‌تر از نظر محاسباتی و حافظه آموزش داد، بدون نیاز به تغییر یا آموزش تمام وزن‌های مدل اصلی. این روش به‌ویژه برای تنظیم دقیق مدل‌های بزرگ در شرایطی که منابع سخت‌افزاری محدود است، بسیار مؤثر و کاربردی است.

هنگامی که از LORA (Low-Rank Adaptation) در آموزش مدل استفاده می کنیم، انتخاب اندازه batch size به نحوی که از همان ابتدا حافظه GPU به حداکثر نر سد، اهمیت زیادی دارد. دلیل این امر این ا ست که در طول فرایند آموزش، م صرف حافظه GPU به تدریج افزایش مییابد و اگر از ابتدا حافظه به حداکثر بر سد، در مراحل بعدی آموزش با م شکل کمبود حافظه (CUDA out of memory) مواجه خواهیم شد.

برای جلوگیری از این مشکل، باید اندازه batch size را به گونهای تنظیم کنیم که در مراحل اولیه آموزش، فضای کافی در حافظه GPU باقی بماند تا بتواند افزایش مصرف حافظه در مراحل بعدی را مدیریت کند. این رویکرد به ما اجازه میدهد تااز حافظه GPU بهینه‌تر استفاده کنیم و احتمال برخورد با خطای کمبود حافظه را کاهش دهیم.

با بررسیهای انجام شده، در این بخش مقدار batch size را برای دادههای آموزشی برابر با ۴ تعیین کردیم.

نتایج آموزش مدل

به دلیل محدودیت‌های سخت‌افزاری موجود در Google Colab، امکان آموزش طولانی مدت مدل فراهم نبود و در نتیجه، فرآیند آموزش از ۵ به ۲ epoch محدود شد.

کاهش خطا از epoch اول به epoch دوم نشان‌دهنده بهبود نسبی در عملکرد مدل بود. این کاهش خطا نشان می‌دهد که مدل در حال یادگیری و بهبود توصیف تصاویر است. با این حال، به دلیل محدودیت در تعداد epochها، نتوانستیم به دقت بهینه دست یابیم.

محدودیت اصـــلی در این فرایند، کمبود منابع ســـخت‌افزاری در Google Colab بود. این محدودیت‌ها شامل موارد زیر می‌شود:

حافظه محدود GPU: با توجه به مصـرف بالای حافظه در حین آموزش مدلهای بزرگ، نتوانســتیم تعداد بیشتری از epochها را اجرا کنیم.

محدودیت زمانی: زمان اختصاص داده شده به هر جلسه در Google Colab محدود است و این موضوع مانع از اجرای جلسات طولانی‌تر برای آموزش مدل می‌شود.

نتایج معیارهای ارزیابی مدل در تصویر ۱۲ قابل نمایش است.

```
Evaluating: 100%|██████████| 125/125 [52:48<00:00, 25.34s/it]
{'BLEU': 0, 'CIDEr': 8.438937706617168, 'ROUGE': 0.9421051070796131, 'METEOR': 0.7349232791607146}
```

تصویر ۱۲ – نتایج ارزیابی مدل بعد از Fine Tune مدل LoRA

تحلیل مقادیر معیارهای ارزیابی

BLEU: 0

این مقدار صفر نشان می‌دهد که هیچ کدام از کپشن‌های تولید شده توسط مدل با کپشن‌های مرجع تطابق ندارند.

CIDEr: 8.438937706617168

این مقدار نشان دهنده تطابق بالا بین کپشن‌های تولید شده توسط مدل وکپشن‌های مرجع است.
CIDEr بر اساس تکرار کلمات و عبارات در کپشن‌های مرجع عمل می‌کند و مقدار بالا نشان‌دهنده

تولید کپشن‌های دقیق است.
مدل در تولید کپشن‌هایی که با کپشنهای مرجع مشابهت دارند، عملکرد خوبی داشته است.

ROUGE: 0.9421051070796131

مقدار بسیار بالایی است که نشان‌دهنده توانایی مدل در بازتولید بخش‌های کلیدی متن مرجع است.
مدل در بازشناسی و تولید عبارات مهم و کلیدی عملکرد بسیار خوبی دارد.

METEOR: 0.7349232791607146

این مقدار نشان دهنده تطابق خوب بین کپشن‌های تولید شده توسط مدل وکپشن های مرجع است.
METEOR بر اساس تطابق کلمات، استمینگ و مترادف‌ها عمل می‌کند.
مدل در تولید کپشن‌هایی که از نظر معنا و مترادف‌ها با کپشن‌های مرجع مشابهت دارند، عملکرد خوبی دارد.

نتایج به دست آمده نشان می‌دهند که مدل در معیارهای ROUGE،CIDEr و METEOR عملکرد خوبی دارد، اما در معیار BLEU بسیار ضعیف عمل کرده است.

در واقع بین مقادیر BLEU و CIDEr ممکن است تناقض به نظر برسد، اما این مسئله می‌تواند ناشی از تفاوت‌های اساسی بین این دو معیار باشد.

BLEU بیش‌تر بر روی تطابق نواحی خاص[1]بین پیشبینی‌ها و کپشن‌های مرجع تمرکز دارد و معمولاً حساس به ترتیب کلمات وتطابق دقیقی از n-grams است. مقدار BLEU پایین می تواند به دلیل عدم تطابق دقیق کلمات یا عبارات در پیشبینیها با کپشنهای مرجع باشد. ولی CIDEr براساس تکرار کلمات و عبارات کلیدی در کپشن های مرجع عمل می‌کند و بیشتر به تشابه معنایی توجه دارد و برای ارزیابی کیفیت توصیف‌های تصویری مناسب است و ممکن است نسبت به تفاوتهای جزئی کلمات حساس نباشد.

[1] n-grams

این به این معناست که اگر مدل توانسته باشد مفاهیم کلی را به درستی توصیف کند، ممکن است CIDEr بالا باشد حتی اگر تطابق دقیقی در n-grams مشاهده نشود. پس اگر مدل قادر به تولید توصیف‌هایی باشد که از نظر معنایی درست هستند ولی با n-grams دقیقاً مشابه کپشن‌های مرجع مطابقت ندارند، CIDEr ممکن است بالا باشد در حالی که BLEU پایین است.

منابع و مراجع

[1] https://zenodo.org/records/10821435

[2] https://huggingface.co/docs/transformers/model_doc/vit_mae

[3] https://huggingface.co/google/vit-base-patch16-224-in21k

Abstract

In this exercise, the multimedia dataset ROCOv2, which includes radiology images along with related medical concepts and captions, was examined. This dataset, extracted from the open-access subset of PubMed, contains seven different clinical modalities, and its concepts were manually collected and evaluated by a radiology specialist.

Next, the pre-trained ViTMAE model from the Transformer library was loaded, and its performance in reconstructing images from the Test set with 75% masking was evaluated. Then, this model was fine-tuned on the Train data specific to the medical imaging domain, and the reconstruction of Test image samples after retraining was examined.

In the third section, using the LoRA method and the PEFT library, a multimodal model was built and fine-tuned to generate captions for medical images from the ROCOv2 dataset. The final model's performance was assessed using standard image captioning metrics such as CIDEr, BLEU, METEOR, and ROUGE. The results demonstrated the capability of these models in analyzing and reconstructing images as well as generating relevant medical captions.

Code link:

https://colab.research.google.com/drive/1kJd_EklY0U3rknEpLZq4Ly_JcZbXjsxB?usp=sharing

Key Words: Masked Autoencoder, Multi Modal, Image Captioning, LoRA

Amirkabir University of Technology
(Tehran Polytechnic)

Department of Mathematics and Computer science

Homework 3
Masked Autoencoder and Multi Modal

By
Mohammadreza Shahrestani
Hossein Khamooshi

Supervisor
Dr. Saeed Sharifian

July 2025