



دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیووتر  
پروژه درس پردازش زبان‌های طبیعی

عنوان:

کلیپ برای تصاویر هیستوپاتولوژی و گزارش‌های آن  
CLIPPath (CLIP for pathology images and reports)

اعضای گروه:

محمدحسین موثقی‌نیا - رضا عباسی - حسن علیخانی - علی سلمانی - حسین  
جعفری‌نیا - مهدی شادرودی

استاد راهنما:

دکتر احسان‌الدین عسگری

۱۴۰۱ اسفند

## چکیده

ابتدا تمامی pdf‌های گزارش‌های آسیب شناسی<sup>۱</sup> مربوط به سرطان پستان<sup>۲</sup> از سایت TCGA<sup>۳</sup> دانلود شده سپس گزارش‌هایی که بخش Diagnosis دارند و بخش آن‌ها هم چندین بخش مختلف دارد انتخاب شد. از این بخش گزارش‌ها، چند جمله ارزشمند از طریق با استفاده هم‌فکری دو نفر از اعضای تیم به صورت دستی استخراج شد. در نهایت تصاویر WSI<sup>۴</sup> متناظر این گزارش‌ها و قسمت انتخاب شده از متن به عنوان داده مورد استفاده قرار گرفت. دو مدل مختلف یادگیری، یکی به عنوان مدل پایه که همان ساختار اصلی کلیپ<sup>۵</sup> را داشت و یک مدل دیگر که به عنوان مدل اصلی استفاده شد که با استفاده از یک ساختار متفاوت تصاویر ارزشمند انتخاب شده و به کمک آن آموزش دیده است، مورد استفاده قرار گرفت. همانطور که در نتایج آورده شده است، مدل اصلی دقت بهتری کسب کرده است اما همچنان جای کار بیشتر و استفاده از ایده‌های بهتر در این زمینه وجود دارد.

## ۱ مقدمه

### ۱-۱ یادگیری خود-نظرارتی

یادگیری خود-نظرارتی<sup>۶</sup> یکی از زیرشاخه‌های یادگیری ماشین است که در آن می‌توان مدل‌ها را بدون نیاز به برچسب آموزش داد. فرایند آموزش در مدل‌های خود-نظرارتی شامل دو بخش است. مرحله اول pretext task است که در آن سعی می‌کنیم با تعیین یک وظیفه خلاقانه، مثل تشخیص میزان چرخش تصویر<sup>[۱]</sup>، بازتولید تصویر بهم‌ریخته و یا مخدوش شده<sup>[۲]</sup>، رنگ کردن تصویر<sup>[۳]</sup> و... مدل را آموزش دهیم. در واقع در این فرایند که مدل نیاز است تا یکی از کارهای تعیین شده را انجام دهد، برای آنکه به عملکرد خوب برسد، باید بتواند ویژگی‌های خوبی از تصویر استخراج کند. همین باعث می‌شود مدل ما در انتهای آموزش به توانایی استخراج ویژگی‌ها از تصاویر برسد بدون آن که لازم باشد از برچسب‌ها تصویر استفاده شود. مرحله دوم downstream task است که در آن، مدلی که به کمک pretext task<sup>task</sup> آموزش دادیم، را برای یک کار خاص مثل کلاس‌بندی، به کمک داده برچسب‌دار آموزش می‌دهیم. تفاوتی که این حالت با حالتی که تنها نیاز است که مدل فقط به شکل بانظار<sup>۷</sup> روی تصاویر آموزش ببینید این است که در اینجا به داده‌های

<sup>1</sup>histopathology

<sup>2</sup>BRCA ( Breast Cancer project)

<sup>3</sup>The Cancer Genome Atlas Program

<sup>4</sup>Whole Slide Image

<sup>5</sup>CLIP

<sup>6</sup>Self-Supervised Learning

<sup>7</sup>Supervised

برچسب دار کمتری نیاز داریم تا به دقت های بالا برسیم؛ زیرا در مرحله قبل مدل به توانایی های خوبی دست پیدا کرده است و لازم نیست با حجم بالای داده، دوباره فرایند یادگیری را تکرار کند.

## ۲-۱ مدل DINO

اواخر سال ۲۰۲۱ گروه تحقیقاتی شرکت فیسبوک<sup>۱</sup> مدلی را تحت عنوان DINO معرفی کرد[۴]. این مدل خود نظارتی در بسیاری از وظایف از جمله بازیابی تصویر<sup>۲</sup>، قطعه بندي تصویر<sup>۳</sup> و دسته بندي تصاویر<sup>۴</sup> توانست بهترین مدل<sup>۵</sup> زمان معرفی خود باشد. نویسندهای این کار در متن مقاله عنوان کرده اند که با انتشار مدل های ترنسفورمری ViT انتظار می رفت که به میزانی که مدل های ترنسفورمری باعث ایجاد تحول در پردازش متن شدند، به همان میزان نیز در پردازش تصویر به تحول دست پیدا کنیم؛ ولی این اتفاق رخ نداد. پس از بررسی های متعدد، این گروه متوجه می شوند که علت موفقیت مدل های ترنسفورمری مثل BERT در پردازش متن، ترکیب مدل های ترنسفورمری و آموزش خود نظارتی است که باعث می شود مدل، چنان ویژگی های مثبتی پیدا کند که نسبت به سایر رقبای خود، با اختلاف بهترین باشد. یادگیری خود نظارتی در کنار استفاده از مدل های ترنسفورمری راز موفقیت مدل DINO است. در واقع آنان نشان دادند که وقتی مدل ViT را به صورت خود-نظارتی آموزش می دهیم، خروجی مدل، امبدینگ های بسیار با کیفیتی است که می تواند در بسیاری از وظایف بهترین نتایج برسد. قبل از آن که وارد معرفی سایر بخش ها شویم توضیحاتی را در مورد نحوه آموزش مدل DINO می دهیم. همان طور که از تصویر نیز قابل مشاهده است، مدل DINO از دو شبکه شاگرد<sup>۶</sup> و استاد<sup>۷</sup> تشکیل می شود. فرایند آموزش به این صورت است که ابتدا روی یک تصویر دو مدل augmentation متفاوت اعمال می شود. این augmentation می تواند مواردی مثل global crop<sup>۸</sup>، random flip<sup>۹</sup>، random crop<sup>۱۰</sup>، color jitter<sup>۱۱</sup> و ... باشد. تنها نکته ای که باید حتماً لحاظ شود این است که شبکه استاد از global crop<sup>۸</sup> تغذیه می کند و شبکه شاگرد از local crop<sup>۱۰</sup>. منظور از local crop<sup>۱۰</sup> این است که برش هایی است که بیش از ۵۰ درصد تصویر اصلی را در برمی گیرند و منظور از global crop<sup>۸</sup> برش هایی است که کمتر از ۵۰ درصد تصویر را در برمی گیرند. بعد از آن که این دو دسته تصویر به شبکه ها داده می شود در انتهای امبدینگ هایی برای هر کدام به دست می آید. مدل باید تلاش کند که امبدینگ خروجی هر دو مدل collapse<sup>۱۲</sup> شبهیه به هم باشد؛ زیرا از یک تصویر به دست آمده اند. مازول centering<sup>۱۳</sup> نیز برای جلوگیری از

<sup>1</sup>facebook

<sup>2</sup>image retrieval

<sup>3</sup>image segmentation

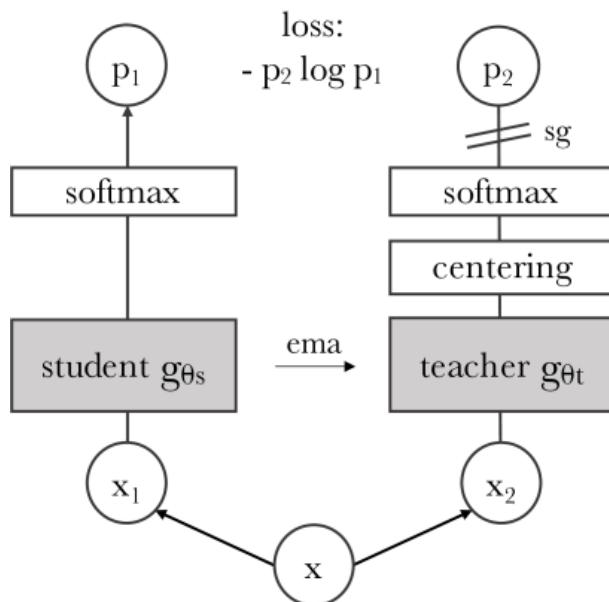
<sup>4</sup>image classification

<sup>5</sup>state-of-the-art

<sup>6</sup>student

<sup>7</sup>teacher

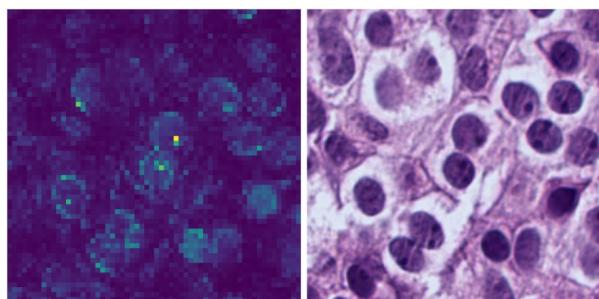
در مدل اعمال شده است. collapse در واقع به حالتی گفته می‌شود که مدل تلاش می‌کند با فریب دادن ما، میزان تابع هزینه خود را کم کند. برای مثال می‌تواند مستقل از این که تصویر ورودی چه تصویری باشد، دو امبدینگ ثابت خروجی بدهد که این باعث می‌شود میزان شباهت آنها حداکثری باشد و میزان هزینه نزدیک صفر شود. تعداد زیادی از مدل‌های یادگیری خود-نظرارتی دیگر از تکنیک negative sample برای آموزش استفاده می‌کنند [۵]. تکنیکی که در آن مدل باید علاوه بر اینکه تلاش کند که امبدینگ‌های خروجی شبکه‌ها به هم نزدیک باشند، تلاش می‌کند که از امبدینگ سایر تصاویر موجود در batch دور باشد که همین باعث می‌شود مدل نتواند فریبی که بالا ذکر کردیم را اجرایی کند. ولی چون مدل DINO از این تکنیک استفاده نمی‌کند، باید به طریقی دیگر با این مشکل روبرو شود که در اینجا از centering استفاده می‌کند.



شکل ۱: نحوه تعریف تابع هزینه برای مدل DINO

حالا ما در مورد مدل DINO توضیحاتی دادیم. در ادامه می‌خواهیم در مورد کاربرد این مدل در مسئله خودمان صحبت کنیم. در بخشی از فاز دوم پژوهش ما قصد داریم که برخی patch‌های مهم تصویر را نگه داریم و آن مواردی که اطلاع خاصی در خود ندارند را حذف کنیم تا به این صورت داده باکیفیت‌تری داشته باشیم. از آنجاکه هم حجم داده‌ها بسیار بالاست و هم دانش کافی در زمینه تشخیص patch‌های باکیفیت‌تری را نداریم، سعی می‌کنیم از مدل‌های آموزش‌دیده روی تصاویر پاتولوژی استفاده کنیم. یکی از مزیت‌های مدل DINO این است که می‌توانیم از آن به راحتی خروجی attention map بگیریم. این خروجی به ما نشان می‌دهد که مدل به کدام بخش تصویر دقت بیشتری دارد. مدل DINO برای این که بتواند عملکرد معقولی داشته باشد، باید به ویژگی‌های مهمی در تصویر دقت کند. ما قصد داریم که به کمک مدل DINO

فیلتری روی همه تصاویر اعمال کنیم و تنها تصاویری که ویژگی‌های زیادی دارند را نگه داشته و بقیه را کنار بگذاریم. برای این کار بر روی مدل DINO که قبلاً با داده TCGA breast آموزش دیده بود<sup>[6]</sup>، یک مرتبه به تعداد ۵۰ ایپک<sup>۱</sup> مدل را به صورت خود-نظرارتی آموزش می‌دهیم. سپس برای تک‌تک تصاویر موجود، آرایه‌های attention map<sup>۲</sup> را مربوط به لایه‌های مختلف مدل است. را استخراج می‌کنیم. مقادیر موجود در آرایه‌های attention به صورت مقادیری بین ۰ و ۱ هستند. ما برای اعمال فیلتر به این صورت عمل کردیم که اگر در آرایه‌های attention یک تصویر بیشتر از ۷۰ درصد مقادیر موجود در آرایه attention آنها مقداری بالاتر از یک حد آستانه داشته باشند، آنها را به عنوان تصاویر ارزشمند در نظر می‌گیریم. شکل ۲ یک مثال برای این دست موارد است.



شکل ۲: نمونه‌ای از تصویر ارزشمند

### ۳-۱ امبدینگ متن

پس از ظهرور مدل‌های زبانی ماسک شده مانند BERT شاهد پیشرفت‌های زیادی در حوزه پردازش زبان‌های طبیعی بوده‌ایم، به‌طور خاص مطالعات زیادی بر روی توسعه مدل‌های سیاق محور<sup>۳</sup> در دامنه‌های مختلف از جمله کلینیکی، پزشکی و زیست-پزشکی انجام شده است [۷، ۸، ۹، ۱۰]. یک روش مرسوم در مطالعات یادشده آموزش مجدد<sup>۴</sup> مدل‌های زبانی عمومی<sup>۵</sup> برای یک دامنه خاص از لغات و جملات است. با این وجود در مواردی مانند پاتولوژی که اصطلاحات تخصصی بسیاری دارند مدل‌های مبتنی برای آموزش مجدد عملکرد خوبی نداشته و اغلب خروجی‌های مرتبطی را تولید نمی‌کنند. یکی از دلایل عدم موفقیت این مدل‌ها، استفاده از مجموعه‌ای از پیش تعیین شده Word-Piece<sup>۶</sup>ها برای توکن‌بندی<sup>۷</sup> بدون نظارت<sup>۸</sup> است [۱۱]. مجموعه لغات در این روش شامل مجموعه‌ای از لغات یا زیرلغات پراستفاده است و هر لغت جدید به صورت

<sup>1</sup>epoch

<sup>2</sup>contextualized

<sup>3</sup>re-train

<sup>4</sup>general-domin

<sup>5</sup>tokenization

<sup>6</sup>unsupervised

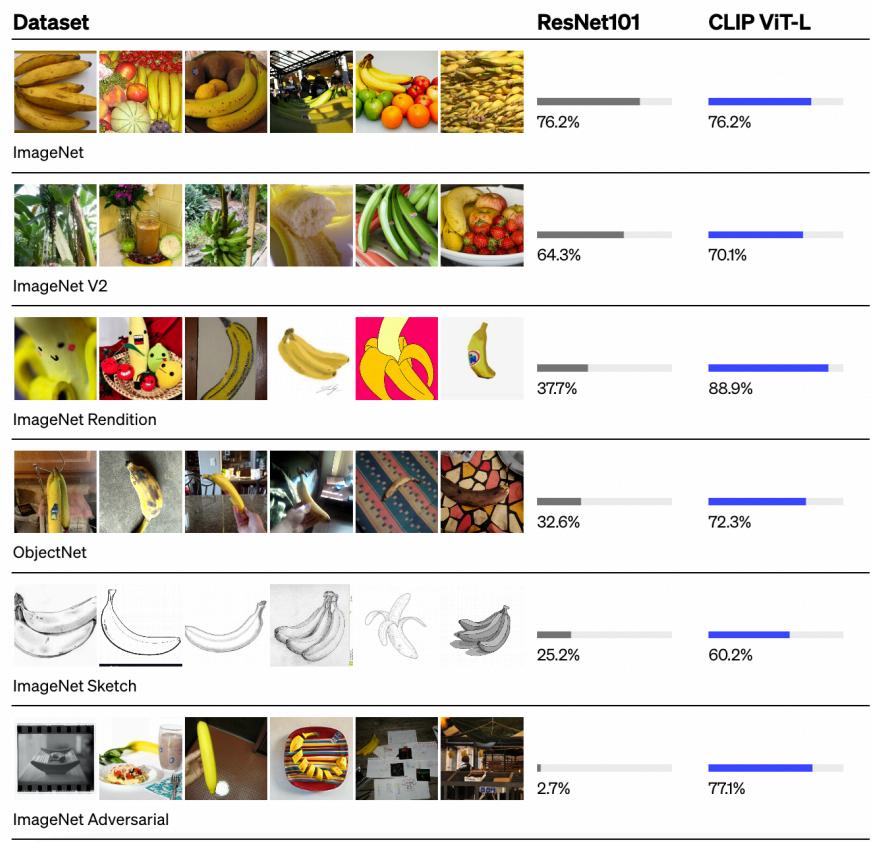
ترکیبی از اعضای این مجموعه توکنایز می‌شود. مدل‌های فوق که این روش توکن‌بندی هستند، در مواجهه با کلمات تخصصی دامنه پاتولوژی مانند carcinoma به اشتباه این لغت را به توکن‌های [’car’, ’##cin’, ’##noma’] می‌شکند که باعث از بین رفتن معنی واقعی کلمه می‌شود. این گونه توکن‌بندی در مدل زبانی باعث نمایش نادرست سیاق محور لغات و جملات شده و دقت مدل در تحلیل‌های بعدی را تحت تأثیر قرار می‌دهد. به منظور رفع مشکلات مذکور، مدل زبانی pathologyBert [۱۲] ارائه شد که علاوه بر مدل زبانی، توکنایزر متناسب با آن نیز بر روی مجموعه‌ای از گزارش‌های پاتولوژی آموزش‌دیده است و در دامنه متون مرتبط با پاتولوژی به خوبی عمل می‌کند.

## ۴-۱ مدل CLIP

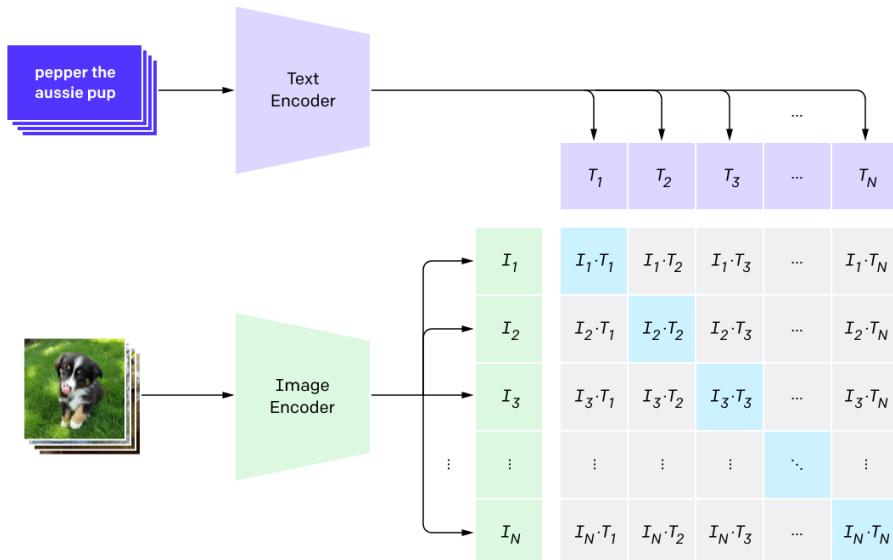
با وجود آنکه یادگیری عمیق و شبکه‌های CNN و ViT تحول عظیمی را در حوزه بینایی ماشین ایجاد کرده است؛ ولی هنوز یکسری چالش اساسی وجود دارد. چالش اول در این مورد است که تهیه دیتاست‌هایی با برچسب‌گذاری‌های دقیق به شدت هزینه دارد و تنها تعداد محدودی دیتاست عظیم به این سبک وجود دارد. مورد بعدی در مورد خود مدل‌هاست. مدل‌هایی که معرفی شده‌اند اکثراً تنها بر روی یک وظیفه و تنها روی دیتاستی که با آن آموزش‌دیده‌اند خوب کار می‌کنند و روی دیتاست‌های دیگر خوب کار نمی‌کنند. شرکت OPEN AI با معرفی مدل CLIP [۱۳] به صورت همزمان این دو مشکل را حل کرده است. توانایی در آموزش از روی تصاویر به کمک نظارت متن باعث شده که بتوان این مدل را از روی داده‌هایی که به راحتی از سطح اینترنت قابل استخراج است آموزش داد و دیگر نگران برچسب تصاویر نبود چرا که به راحتی می‌توان یک متن که احتمالاً متن مربوط به تصویر است را از اینترنت استخراج کرد و با آن مدل را آموزش داد. از طرفی مدل معرفی شده بر روی تعداد زیادی از دیتاست‌ها و وظایف مختلف نتایج بسیاری عالی می‌گیرد. این نتایج خوب به خصوص در ارزیابی zero-shot خیره‌کننده است. به عنوان یک مثال نتایج مقایسه مدل CLIP با مدل ResNet<sup>۱۴</sup> که روی imageNet آموزش‌دیده است در شکل ۳ آمده که همان‌طور که مشاهده می‌کنیم نتایج در ارزیابی روی imageNet-R از ۷۶ درصد تفاوت دقت را نشان می‌دهد.

این مدل الهام گرفته از مقاله Learning Visual N-Grams from Web Data است که نشان داده‌اند نظارت از طریق زبان می‌تواند توانایی مدل در zero-shot transfer را افزایش دهد [۱۴]. نحوه آموزش مدل CLIP در شکل ۴ آمده است.

مدل به این صورت عمل می‌کند که تعدادی جفت تصویر و متن را به عنوان ورودی می‌گیرد. مدل از دو شبکه انکودر تصویر و انکودر متن تشکیل شده است. بعد از آن که به کمک این دو شبکه، امبدینگ آنها را به دست آورد، سعی می‌کند به کمکتابع contrastive loss که دارد، امبدینگ تصویر به امبدینگ متن آن نزدیک و از سایر متون دور باشد. برای امبدینگ متن هم به همین صورت عمل می‌کند.



شكل ٣: مقایسه مدل CLIP و مدل ResNet



شكل ٤: نحوه آموزش مدل CLIP

## ۲ روش

### ۱-۲ پیش‌پردازش متن

ابتدا تمامی pdf‌های گزارش آسیب‌شناسی<sup>۱</sup> مربوط به سرطان پستان<sup>۲</sup> از سایت TCGA [۱۵] با استفاده از نرم‌افزار فراهم‌شده توسط خودشان دانلود شده (۱۱۰۰ فایل) سپس گزارش‌هایی که بخش Diagnosis دارند و بخش آن‌ها هم چندین زیربخش مختلف دارد، انتخاب شد. علت این کار این است که گزارش‌های آسیب‌شناسی ساختار یکسانی ندارند و بعضی به صورت متنی و بعضی به صورت جدول هستند، و بعضی به توضیحاتی کم و بعضی به توضیحات بسیار، بسیله کردند. در نتیجه ما که برای آموزش مناسب مدل نیاز به ساختار یکسان داریم از این ساختار که درصد بیشتری از pdf‌ها را تشکیل می‌داد استفاده کردیم. از بین این لیست از pdf‌ها پنجاه نمونه با استفاده از توزیع تصادفی یکنواخت گرفته شد. از بین پنجاه نمونه انتخاب شده، وضعیت تهاجمی بودن یا نبودن milk duct و milk lobule به همراه توضیحات مرتبط با هر کدام انتخاب شد. به طور کلی در این زمینه سرطان پستان می‌تواند حالات زیر را داشته باشد:

• DCIS: حضور سلول‌های غیرعادی در milk duct (Ductal Carcinoma In Situ)

• LCIS: حضور سلول‌های غیرعادی در milk lobule (Lobular Carcinoma In Situ)

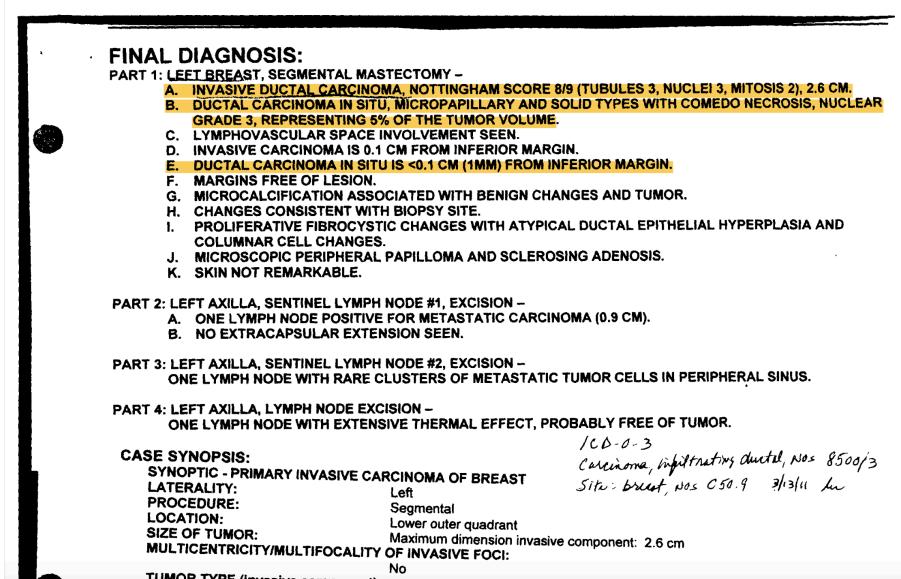
• IDC: حضور سلول‌های تهاجمی در milk duct (Invasive Ductal Carcinoma)

• ILC: حضور سلول‌های تهاجمی در milk lobule (Invasive Lobular Carcinoma)

بخش گفته شده از هر پنجاه pdf توسط دو نفر از اعضای گروه استخراج شد و در یک جلسه تک‌تک موارد توسط هر دو فرد مورد بررسی قرار گرفت و اختلاف‌نظر رفع و یک متن واحد برای هر گزارش استخراج شد. برای مثال قسمتی از یکی از گزارش‌ها در شکل ۵ آورده شده است؛ قسمتی که هایلایت شده، قسمتی است که متن نهایی از آن استخراج و خلاصه شده است. متن استخراج شده مطابق شکل ۶ می‌باشد که حاصل جلسه حل تعارض‌ها است.

<sup>1</sup>histopathology

<sup>2</sup>BRCA



شکل ۵: نمونه‌ای از گزارش پاتولوژی

INVASIVE DUCTAL CARCINOMA, DUCTAL CARCINOMA IN SITU, MICROPAPILLARY AND SOLID TYPES WITH COMEDO NECROSIS.  
DUCTAL CARCINOMA IN SITU IS NEAR INFERIOR MARGIN

شکل ۶: نمونه‌ای از متن استخراج شده حاصل از جلسه حل تعارض‌ها برای تصویر ۵

## ۲-۲ پیش‌پردازش تصویر

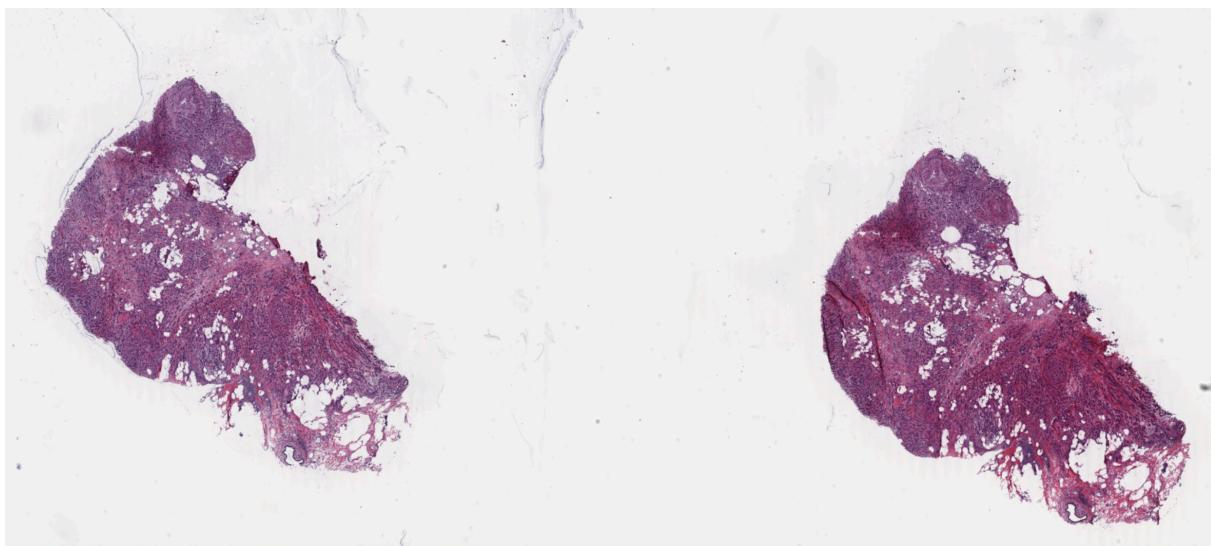
### ۱-۲-۲ دانلود دیتاست

داده‌ی مورد نیاز برای پروژه از دو قسمت متن گزارش و تصاویر مرتبط با آن گزارش، تشکیل شده‌است. برای استفاده و دانلود تصاویر whole slide image، نیاز است UUID هر کدام از تصاویر را پیدا کنیم. این UUID نشانگر یک رشته‌ی منحصر به فرد برای هر فایل است که با استفاده از آن فایل مورد نظر به صورت یکتا از دیتابیس استفاده شده قابل دانلود است. برای این کار manifest کامل دیتاست مورد نظر را از پروژه TCGA-BRCA [۱۶] دانلود می‌کنیم که شامل UUID، نام فایل، سایز فایل و موارد دیگر است. فایل‌های انتخابی زیرمجموعه‌ی این فایل‌ها هستند. از روی نام فایل آن‌هایی که فرمت SVS دارند و جزو گزارش‌های انتخاب شده هستند را فیلتر می‌کنیم. نهایتاً یک فایل manifest خواهیم داشت که تمام موارد آن باید دانلود و پردازش شود.

با توجه به این که تصاویر موردنظر به صورت مستقیم از منبع انتخاب شده قابلیت دانلود شدن را ندارند، نیاز است برنامه‌ی gdc client [۱۷] که پیشنهاد خود منبع است، نصب شود. این برنامه در دو نسخه‌ی colab دارای محیط گرافیکی و بدون محیط گرافیکی ارائه می‌شود که با توجه به فرآیند انجام کار که در

اجرا می شود، نسخه‌ی بدون محیط گرافیکی، نصب شده است. این برنامه UUID ذکر شده در مرحله‌ی قبل را به عنوان ورودی گرفته و فایل مورد نظر را دانلود و ذخیره می‌کند. در colab فایل دانلود شده در root پروژه ذخیره می‌شود. با توجه به فایل manifest ساخته شده، فایل‌ها را یکی یکی دانلود می‌کنیم. تصاویر whole slide image دانلود شده حجمی بین چند ده مگابایت تا چند گیگابایت دارند که در ابعاد تصویر نیز متفاوت هستند. مجموع حجم فایل‌های دانلود شده حدوداً ۶۰ گیگابایت است.

یک نمونه از تصویر WSI دانلود شده - برای بافت سینه- به صورت تغییر اندازه پیدا کرده در شکل ۷ آمده است. تصاویر دانلود شده در نهایت برای استفاده در مدل نهایی باید به فرمت jpg یا png تبدیل شوند. در این روش ما با استفاده از png ذخیره سازی کردیم که کمترین میزان از بین رفتن داده را داشته باشیم.

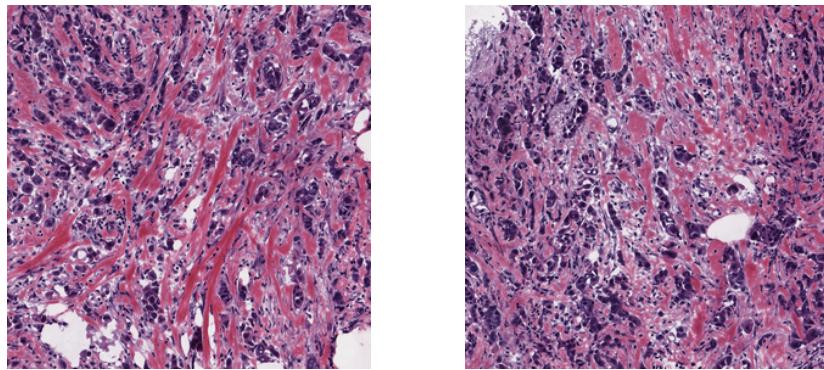


شکل ۷: نمونه‌ای از تصویر WSI بدون هیچ‌گونه پیش‌پردازش، این تصاویر معمولاً ابعاد بسیار بزرگی دارند و در حدود ۲۰ هزار در ۲۰ هزار پیکسل هستند.

## ۲-۲-۲ پچ کردن تصاویر

برای خواندن تصاویر با فرمت svs از کتابخانه openslide استفاده کرده‌ایم. این کتابخانه تابعی را معرفی می‌کند که با فراخوانی آن می‌توانیم بخشی از تصویر را در پایتون برای پردازش کردن، بارگذاری کنیم. امکان بارگذاری کامل تصویر با توجه به حجم زیاد آن و محدودیت حافظه‌ی colab، وجود ندارد.

با توجه به هدف پروژه، نیاز است تصاویر دانلود شده به صورت پچ‌هایی با سایز کمتر تبدیل شوند. برای این کار روی عکس در جهت سط्रی و در جهت ستونی با گام‌های ۱۰۲۴ پیکسلی حرکت می‌کنیم و ناحیه انتخاب شده را با استفاده از openslide می‌خوانیم. با این کار عکس مورد نظر به پچ‌هایی با سایز ۱۰۲۴ در ۱۰۲۴ پیکسل تبدیل می‌شود. از آنجا که همه‌ی این پچ‌ها حاوی شی مورد نظر نیستند و اطلاعات مهمی در خود ندارند و عمدتاً سفید رنگ هستند، نیاز است از مجموعه‌ی پچ‌های انتخاب شده حذف شوند.



شکل ۸: دو نمونه از تکه‌های استخراج شده از تصویر ۷

برای این کار از الگوریتم آماده‌ی otsu استفاده می‌کنیم. این الگوریتم تصویر را به دو دسته‌ی پس زمینه و پیش زمینه، تقسیم می‌کند. اگر درصد زیادی از تصویر مورد نظر به پس زمینه، که سفید است، تعلق داشته باشد، آن را حذف می‌کنیم. این کار را با تعریف یک threshold ثابت انجام می‌دهیم. مشکل این روش در آن است که برای تصاویری که اکثر پیکسل‌های آن سفید رنگ است، کل عکس را به عنوان پیش زمینه در نظر می‌گیرد. برای حل این موضوع یک نسبت دیگری تعریف می‌شود که میزان پیکسل‌های سفید رنگ به کل پیکسل‌های تصویر را حساب می‌کند، اگر پیکسلی در حالت gray scale، مقدار بزرگتر از ۲۲۵ داشته باشد پیکسل سفید حساب می‌شود. اگر برای پچی این نسبت از مقدار مشخصی بالاتر باشد، آن پچ انتخاب نخواهد شد. در نهایت پچ‌هایی باقی خواهد ماند که حاوی اطلاعات مهم هستند. این کار را بار دیگر با گام ۳۰۷۲ و ۴۸۰ پیکسل انجام می‌دهیم.

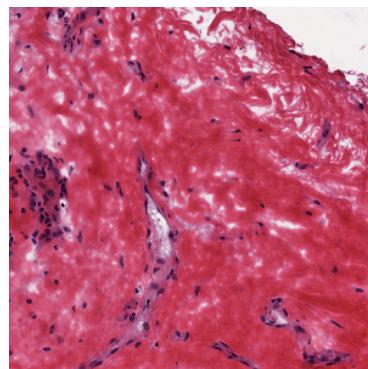
خروجی این بخش برای هر تصویر شامل سه سری پچ‌های با سایزهای ۱۰۲۴، ۲۰۴۸ و ۳۰۷۲ است که در نهایت برای نرمال کردن به سایز ۲۹۹ در ۲۹۹ پیکسل تغییر اندازه داده شده‌اند. در مجموع تمامی پچ‌های انتخاب شده حدود ۵۰ گیگابایت شد که در چهار درایو گوگل مختلف ذخیره سازی شد.

### ۳-۲-۲ نرمال‌سازی پچ‌ها

ابتدا تعدادی محتوای آموزشی که آشنایی ابتدایی در رابطه با بافت ریه و پستان می‌دهند مشاهده شد سپس با استفاده از نرم افزار PMA.start برسی چشمی ۹۰ تصویر WSI تصادفی یکنواخت سالم ریه و ۳۰ تصویر WSI تصادفی یکنواخت سالم پستان جهت آشنایی با فضای بافتی مجموعه داده TCGA انجام شد. در نهایت ۳ تصویر WSI انتخاب شد و قطعه قطعه شده و بهترین آن‌ها که نماینده خوبی از کلیه بافت‌های پستان است و کیفیت لازم برای عکس مرجع نرمال سازی به روش مطرح شده در این مقاله [۱۹] را دارد انتخاب شد. شکل ۹ تصویری است که به عنوان تصویر مرجع مورد استفاده قرار گرفته است. چنین قطعه

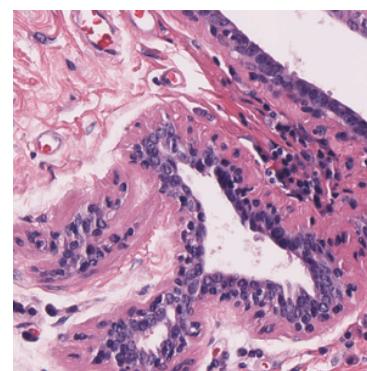
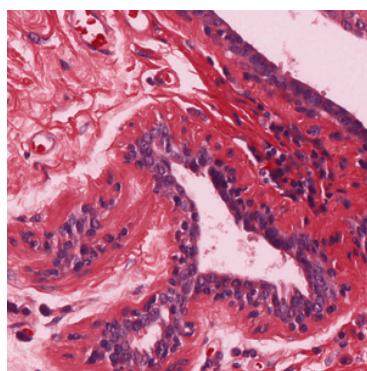
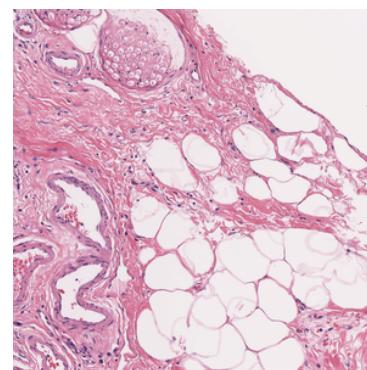
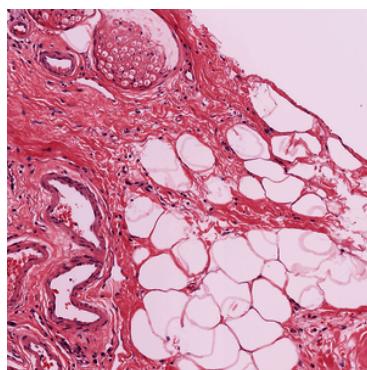
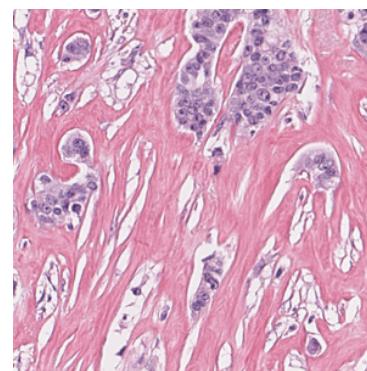
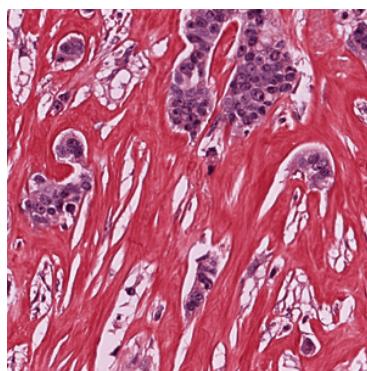
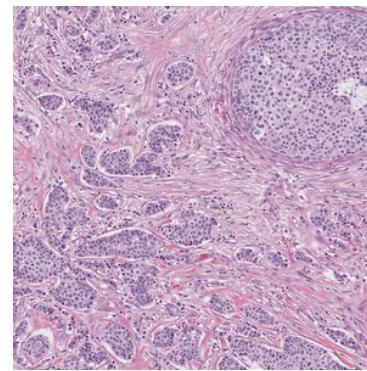
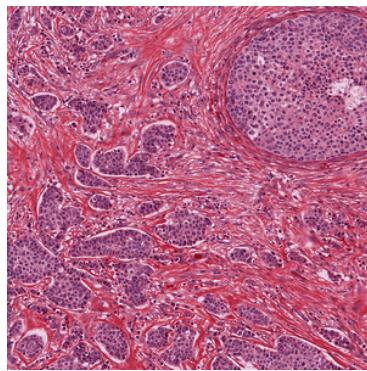
ای باید حاوی ویژگی‌های زیر باشد:

- بخش پس زمینه و سفید کمی داشته باشد.
- بخش‌های مختلف بافت سینه مانند gland و یا بافت میانی را داشته باشد.
- رنگ آن میانگین کیفی از رنگ WSI‌ها باشد.
- تراکم تعداد هسته‌های موجود در آن میانگین چشمی تراکم تعداد هسته‌های موجود در WSI‌ها باشد.



شکل ۹: تصویر مرجع، این تصویر به عنوان تصویر مرجع نرمال‌سازی استفاده شده است.

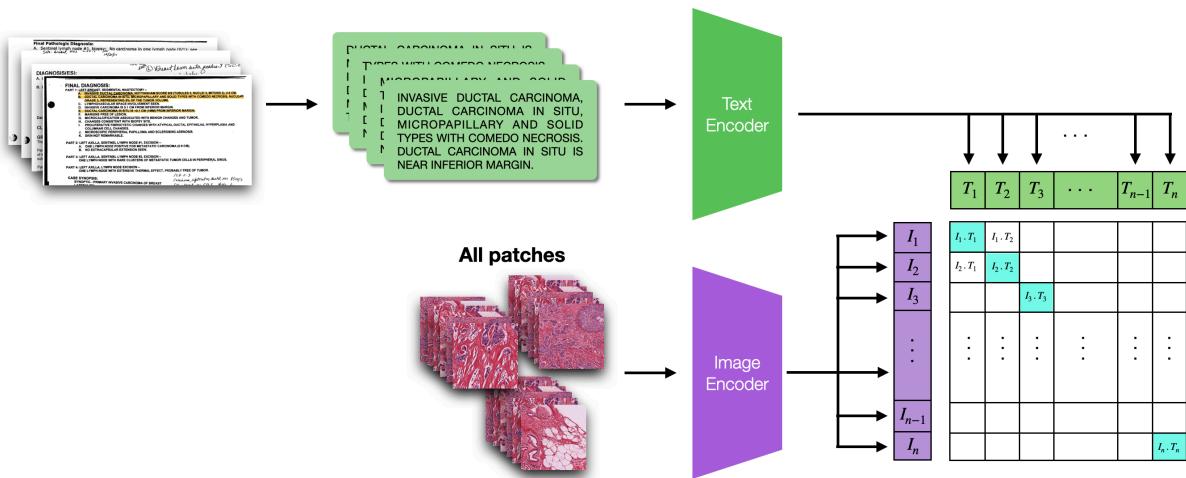
در نهایت نرمال‌سازی رنگی به کمک نرمافزار staintools [۲۰] که مبتنی بر این مقاله [۲۱] پیاده سازی شد است؛ انجام گرفته است. این نرم افزار به این صورت کار می‌کند که یک تصویر را به عنوان مرجع دریافت کرده و متناسب با آن بقیه تصاویر را نرمال می‌کند. این نرمال سازی برای رنگ‌های تصویر است؛ به این معنی که ترکیب رنگی که بعد از نرمال سازی به دست می‌آید بسیار مشابه تصویر مرجع است. هدف از این کار این است که، مدلی که بر روی این تصاویر آموزش می‌بینند، تلاش کند ویژگی‌های اصلی تصویر شامل تغییر شکل‌های مرتبط با هسته سلول‌ها و مواردی از این دست را استخراج کند؛ چرا که اگر این نرمال سازی صورت نگیرد، ممکن است مدل یادگیرنده نسبت به تغییرات رنگ در تصاویر مختلف حساس شده و به اشتباه ویژگی‌هایی غیرمرتبط با وظیفه اصلی یاد بگیرد و هرچند ممکن است نتایج خوبی داشته باشد ولی صرفا تاثیر شباهت این ویژگی غیرمرتبط در کلاس‌های مختلف است و اصطلاحاً به آن تاثیر دسته می‌گویند که به این معنی است که مدل به خوبی ویژگی‌ها را استخراج نکرده است و صرفا ویژگی‌هایی را یادگرفته که کمک کرده تا تصمیم درستی بگیرد و براساس یک ویژگی غیرزیستی موجود در تصاویر بوده است از جمله رنگ و... که به این ترتیب اگر مجموعه دادگان تغییر کند؛ عملکرد مدل به شدت کاهش پیدا می‌کند. به زبانی دیگر مدل به خوبی تعیین پیدا نکرده است. (نمونه‌ای از چند قطعه تصویر نرمال شده و غیر نرمال در شکل ۱۰ آورده شده است).



شکل ۱۰: تصویر نرمال شده و ورودی از پچهای ستون سمت راست: تصویر نرمال شده، ستون سمت چپ: تصویر ورودی

## ۳-۲ مدل پایه

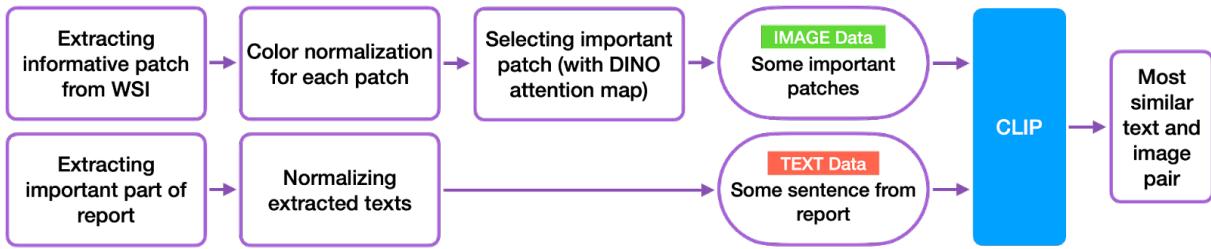
در مدل پایه، تصاویر و متونی که برای این بخش آماده شده است را برای آموزش به مدل می‌دهیم. (شکل ۱۱) در بخش انکودر تصویر از یک مدل ViTB/32 استفاده کردیم و در بخش انکودر متن نیز از یک مدلی شبیه به BERT که شش لایه است استفاده می‌کنیم. تصاویری که به این مدل می‌دهیم، در واقع تکه‌هایی با سایز ۳۰۷۲ در ۳۰۷۲ پیکسل است که به ابعاد ۲۲۴ در ۲۲۴ تغییر سایز داده‌ایم تا بتوان آن را به مدل داد. برای محتوای متنی نیز از گزارش‌هایی مربوط به دیتاست و بخش diagnosis آن‌ها استفاده می‌کنیم. مدل را در حالت from scratch با کارت گرافیک ۳۰۹۰ آموزش می‌دهیم. در این آموزش تعداد ایپک‌ها، ۵۰ در نظر گرفته شده است. Batch size نیز برابر با ۶۴ است و تنها آگمنتیشن که استفاده می‌کنیم CLIP flip است. علت استفاده از این آگمنتیشن به این دلیل است که در مقاله اصلی نیز ذکر شده است که در فرایند آموزش از آگمنتیشن خاصی استفاده نمی‌کنند. پایپلاین کلی در شکل ۱۱ نشان داده شده است.



شکل ۱۱: مدل پایه، صرفا متن‌های پردازش شده و تمامی قطعه‌های استخراج شده از تصویر را به عنوان ورودی می‌گیرد.

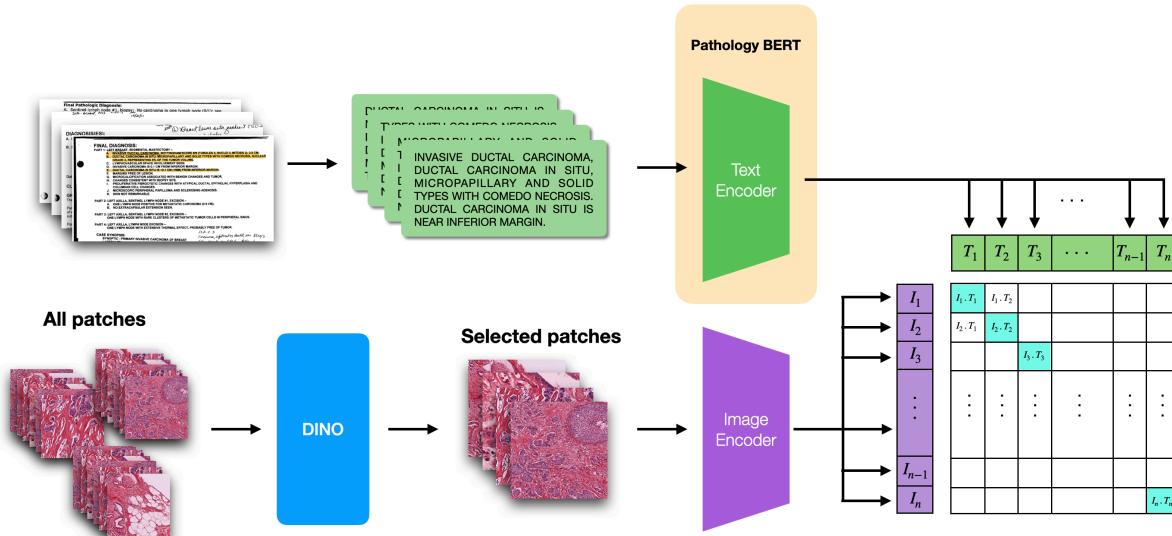
## ۴-۲ مدل اصلی

در مدل معرفی شده سعی می‌کنیم کمی دقیق‌تر عمل کنیم. در این بخش برای آموزش مدل از پچ‌های ۱۰۲۴ در ۱۰۲۴ پیکسلی استفاده می‌کنیم که به ابعاد ۲۲۴ در ۲۲۴ تغییر سایز پیدا کرده‌اند. از آن جا که سایز پچ‌ها نسبت به حالت پایه ای کوچک‌تر است، بنابراین تعداد تصاویر تغییر سایز یافته ما زیاد خواهد بود. برای آن که از تصاویر حاوی اطلاعات بالا استفاده کنیم، همان گونه که در بخش قبل به طور کامل توضیح دادیم، به کمک مدل DINO که pretrain شده روی مجموعه دادگان TCGA و train شده روی داده خودمان است، تصاویری که آنها نشان‌دهنده اطلاعات زیاد نیست را از مجموعه حذف



شکل ۱۲: پایپلاین پردازشی، به صورت کلی فرآیند تحلیل شامل دو بخش پایپلاین پردازش متن که از زیربخش‌های استخراج متن و انتخاب بخش ارزشمند توسط دو نفر از اعضای تیم و سپس نرم‌السازی متن تشکیل شده است. همچنین پایپلاین پردازش تصویر شامل استخراج قطعه‌های ارزشمند از تصویر و نرم‌السازی آن با یک الگوریتم ثابت می‌باشد. پس از این متن و تصویر آماده است و برای مدل پایه از همین دو داده استفاده شده است. اما در مدل اصلی، تصاویر به مدل DINO داده شده و بر روی خروجی آن پردازش‌هایی انجام شده و سپس به مدل اصلی داده شده است.

می‌کنیم و با باقی‌مانده تصاویر مدل را آموزش دهیم. البته ابتدا قصد داشتیم که این فرایند را به صورت end-to-end انجام دهیم؛ ولی به دلیل آن که در مدل CLIP از دو ترنسفورمر بزرگ استفاده می‌شود، ظرفیت اضافه کردن یک ViT جدید که مربوط به مدل DINO است را نداشتیم. در ادامه به جای استفاده از یک pathology bert قبلى، از یک مدل BERT با نام pathology bert استفاده می‌کنیم [۱۲]. این مدل روی بیش از ۳۴۷ هزار گزارش پاتولوژی آموزش‌داده شده است. با جایگذاری این مدل BERT جدید با مدل تکست انکودر قبلى، مدل CLIP را با داده‌های فیلتر شده جدید آموزش می‌دهیم. سایر کانفیگ‌ها مانند مدل پایه است.



شکل ۱۳: مدل ارائه شده، شامل دو بخش پایپلاین زبانی مربوط به پردازش گزارش‌ها و پایپلاین تصویری مربوط به پردازش تصاویر قطعه‌بندی شده می‌باشد. تصاویر ابتدا به مدل DINO داده شده اند و آن‌های که دارای attention map بیشتر از آستانه خاصی باشند، انتخاب شده اند و به عنوان ورودی مدل اصلی استفاده شده اند. همچنین متن‌ها نیز از بخش خاصی از گزارش‌ها از طریق مشورت انتخاب شده اند.

### ۳ نتایج

#### ۱-۳ ارزیابی

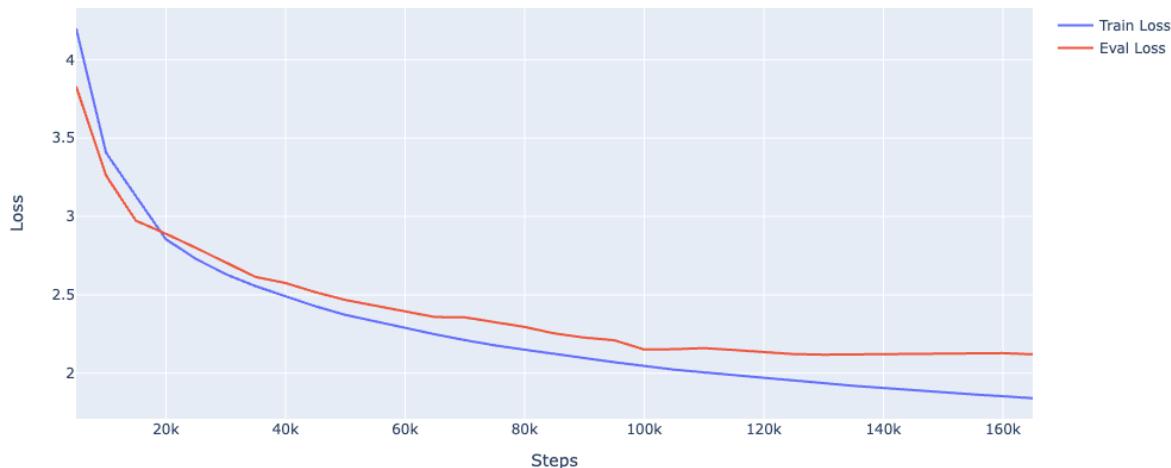
مدل پایه و مدل ارائه شده هرکدام ۵۰ ایپاک آموزش دیده اند و نمودار تغییرات هزینه برای هرکدام از آنها به صورت مجزا در شکل‌های ۱۵ و ۱۴ نشان داده شده است.

در مورد روند ارزیابی به این صورت عمل می‌کنیم که بهازای هر تصویر تست، تعداد مشخصی کپشن که در این جا ۵۰ در نظر گرفته شده است، را به مدل می‌دهیم. به زبانی دیگر یک قطعه از تصویر را به همراه تعداد زیادی متن به مدل می‌دهیم و انتظار داریم که مدل امبدینگی که به عنوان خروجی برای متن‌ها و تصویر می‌دهد به نحوی باشد که امبدینگ متن مرتبط با تصویر کمترین فاصله کسینوسی را از امبدینگ تصویر داشته باشد. اگر مدل در خروجی نشان دهد که نزدیک‌ترین متن به تصویر مدنظر، متن مرتبط با همان تصویر است آن را بهعنوان یک پیش‌بینی درست و در غیر این صورت آن را بهعنوان یک پیش‌بینی غلط در نظر می‌گیریم. در نهایت دقت مدل پایه روی مجموعه داده تستی که برای این مدل در نظر گرفتیم ۵۲.۵ درصد شده است. همچنین برای مدل ارائه شده نیز همین فرآیند انجام شد و دقت آن به ۶۸.۹ درصد رسید. نمونه‌هایی از خروجی مدل ارائه شده و مدل پایه در شکل‌های ۱۶ و ۱۷ پیوست آورده شده است. همچنین دقت ۵ مدل برتر نیز برای مدل پایه و مدل ارائه شده بررسی شد و در جدول ۱ آورده شده است. همان طور که مشاهده می‌شود در مدل ارائه شده دقت، ارتقا پیدا کرده است.

#### ۲-۳ نرم‌افزار تحت وب

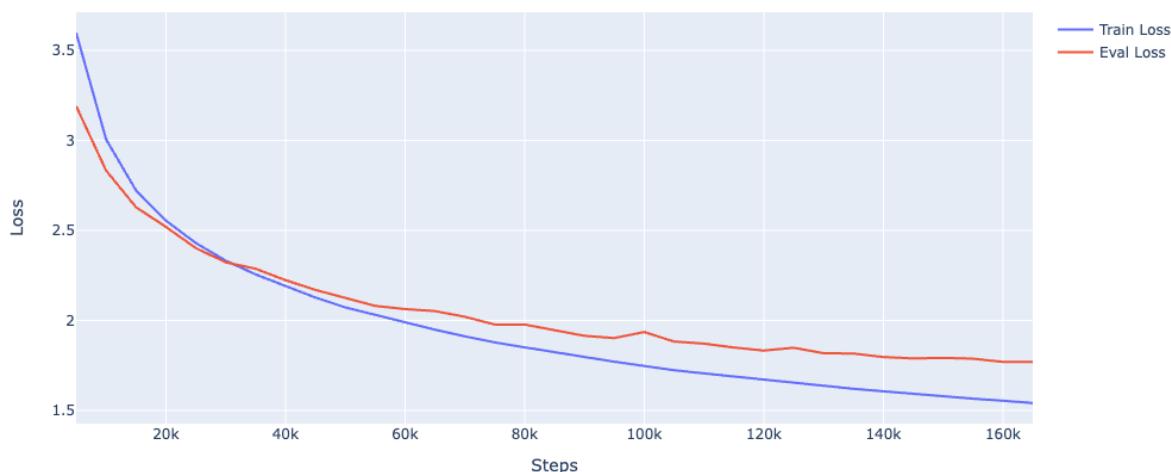
به منظور راه اندازی دمو برخط مدل ایجاده ، از ترکیب فریم ورک های Django و Celery و استفاده شده است. فریمورک Django یک ابزار قدرتمند برای ایجاد نرم‌افزارهای تحت وب مبتنی بر پایتون و Celery ابزاری جهت اجرای دستورات به خط لوله و به صورت ناهمگام است و ترکیب این دو در کنار یکدیگر روش خوبی برای به خدمت گیری مدل های هوش مصنوعی را در اختیار قرار می دهد. وب دمو ایجاد شده روی یک سرور با سیستم عامل لینوکس و ۴ گیگابایت حافظه RAM و ۴ هسته پردازشی با فرکانس ۳ GHz راه اندازی شده است و برای هر تسک جدید به طور متوسط دارای زمان پاسخ ۵ ثانیه می باشد. برای مشاهده جزئیات بیشتر از صفحه مربوط به وب دمو بازدید فرمایید.

Train Loss vs Eval Loss baseline Model



شکل ۱۴: نمودار میزان تغییرات هزینه برای دادگان آموزش و صحت سنجی در زمان آموزش مدل پایه

Train Loss vs Eval Loss proposed Model



شکل ۱۵: نمودار میزان تغییرات هزینه برای دادگان آموزش و صحت سنجی در زمان آموزش مدل ارائه شده

جدول ۱: مقایسه دقت مدل CLIPath و مدل پایه

	Baseline model	CLIPath model
Accuracy	0.525	0.698
Top-5 Accuracy	0.57	0.72

## ۴ بحث

در این پژوهه یک مدل بهبودیافته با نام CLIPPath مبتنی بر CLIP ارائه شده که در مقایسه با مدل پایه که همان مدل CLIP اصلی است، از نظر دقیق‌تر عمل می‌کند. در بخش انکوادر متن این مدل، از pathology استفاده شده که متون حاوی اطلاعات مهم را دریافت می‌کند. پچها نیز قبل از ورود به بخش انکوادر تصویر، از نظر ارزش اطلاعاتی توسط مدل DINO فیلتر می‌شوند و پچهای دارای اطلاعات ارزشمند به انکوادر تصویر داده می‌شوند.

یکی از چالش‌های این پژوهه، محدودیت منابع پردازشی بود که از بهتر آزمودن مدل و همچنین استفاده از ایده‌ها و روش‌های دیگر جلوگیری نمود.

روش‌های بسیار زیادی برای بهبود این مدل وجود دارد که در کارهای آتی می‌توان به آن‌ها پرداخت. برای نمونه می‌توان به انتخاب روش‌های بهتر در انتخاب پچهای با ارزش، طراحی مدل اصلی مبتنی بر روش‌های یادگیری چند نمونه‌ای<sup>۱</sup>، استخراج بخش مهم متن و استفاده از منابع محاسباتی بهتر مثل پردازنده گرافیکی با حافظه بیشتر، اشاره کرد.

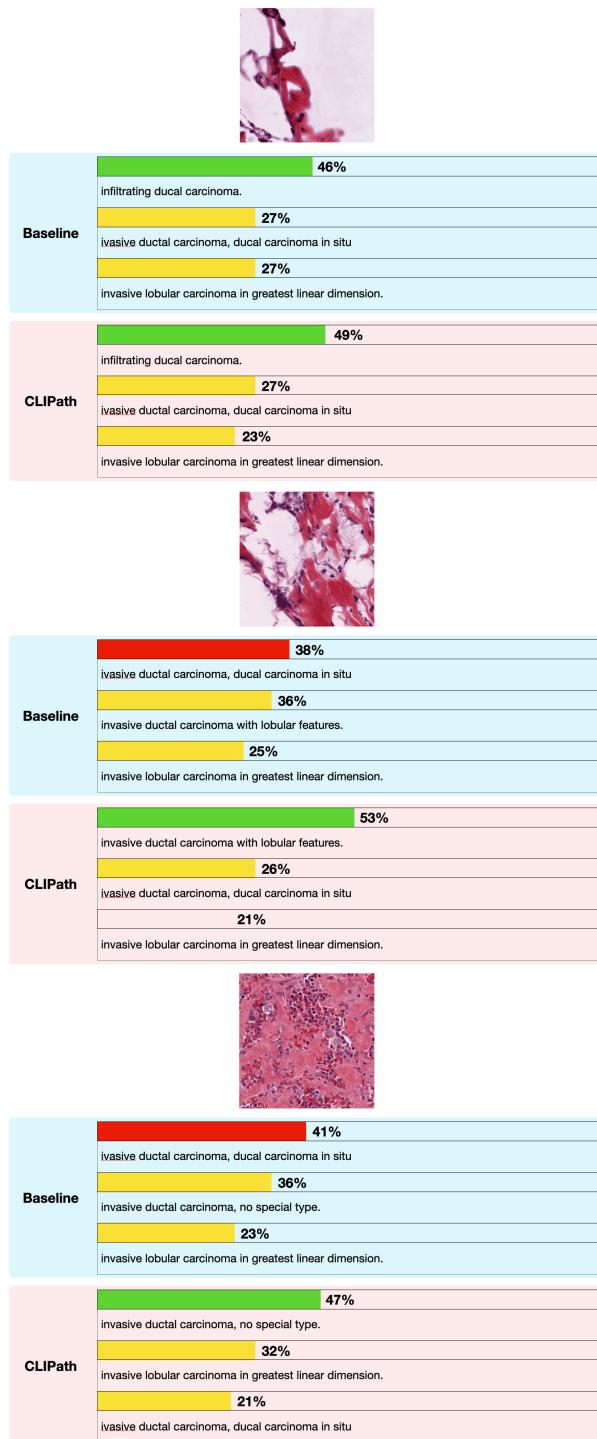
---

<sup>1</sup>Multiple Instance Learning

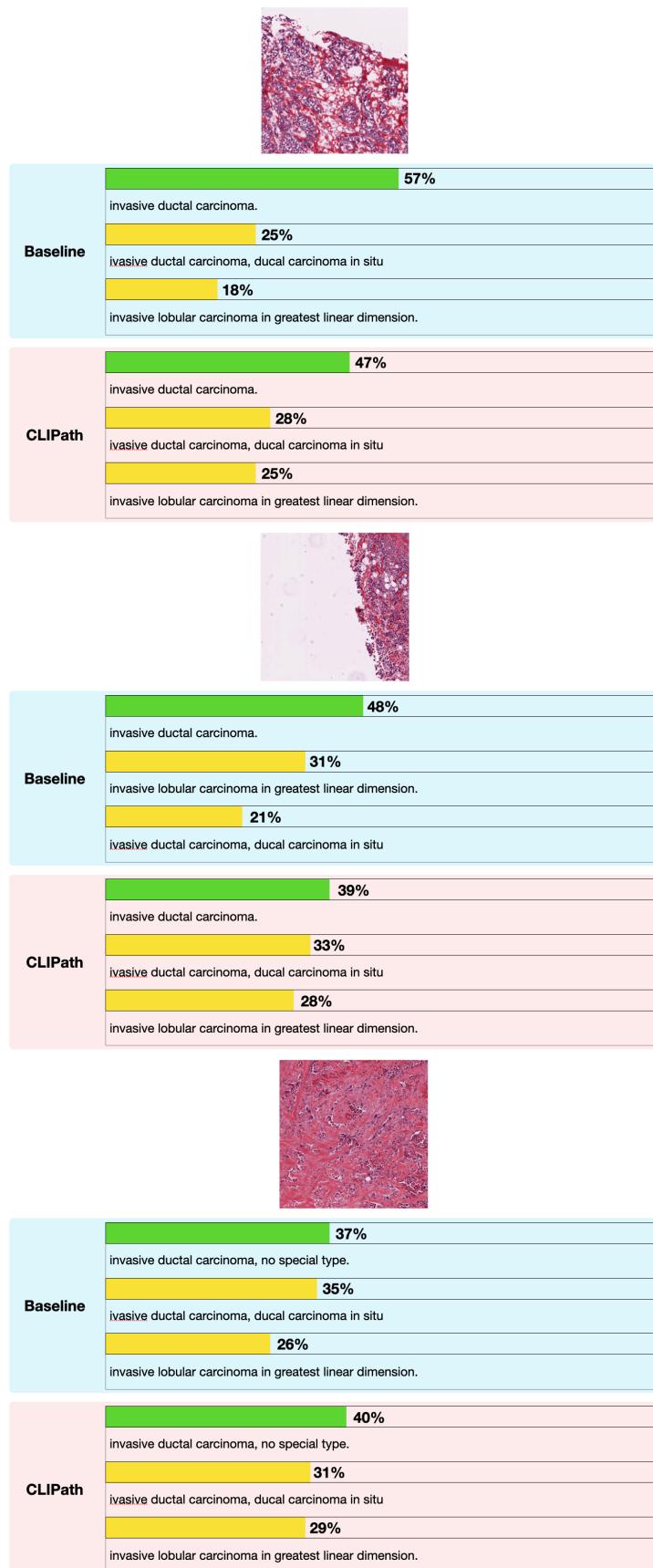
## مراجع

- [1] S. Yamaguchi, S. Kanai, T. Shioda, and S. Takeda, “Image enhanced rotation prediction for self-supervised learning,” in *2021 IEEE International Conference on Image Processing (ICIP)*, pp.489–493, IEEE, 2021.
- [2] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI*, pp.69–84, Springer, 2016.
- [3] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.5505–5514, 2018.
- [4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp.9650–9660, 2021.
- [5] L. Xu, J. Lian, W. X. Zhao, M. Gong, L. Shou, D. Jiang, X. Xie, and J.-R. Wen, “Negative sampling for contrastive representation learning: A review,” *arXiv preprint arXiv:2206.00212*, 2022.
- [6] R. J. Chen and R. G. Krishnan, “Self-supervised vision transformers learn visual concepts in histopathology,” *arXiv preprint arXiv:2203.00585*, 2022.
- [7] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, “Biobert: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol.36, no.4, pp.1234–1240, 2020.
- [8] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, “Publicly available clinical bert embeddings,” *arXiv preprint arXiv:1904.03323*, 2019.
- [9] I. Beltagy, K. Lo, and A. Cohan, “Scibert: A pretrained language model for scientific text,” *arXiv preprint arXiv:1903.10676*, 2019.
- [10] Y. Peng, S. Yan, and Z. Lu, “Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets,” *arXiv preprint arXiv:1906.05474*, 2019.
- [11] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv preprint arXiv:1609.08144*, 2016.

- [12] T. Santos, A. Tariq, S. Das, K. Vayalpati, G. H. Smith, H. Trivedi, and I. Banerjee, “Pathologybert—pre-trained vs. a new transformer language model for pathology domain,” *arXiv preprint arXiv:2205.06885*, 2022.
- [13] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*, pp.8748–8763, PMLR, 2021.
- [14] A. Li, A. Jabri, A. Joulin, and L. Van Der Maaten, “Learning visual n-grams from web data,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp.4183–4192, 2017.
- [15] N. C. Institute, “The cancer genome atlas program (TCGA),” 2023.
- [16] N. C. Institute, “Gdc data portal,” 2023.
- [17] N. C. Institute, “Gdc clinet tool,” 2023.
- [18] A. Goode, B. Gilbert, J. Harkes, D. Jukic, and M. Satyanarayanan, “Openslide: A vendor-neutral software foundation for digital pathology,” *Journal of pathology informatics*, vol.4, no.1, p.27, 2013.
- [19] W. Zhu, C. Fernandez-Granda, and N. Razavian, “Interpretable prediction of lung squamous cell carcinoma recurrence with self-supervised learning,” *arXiv preprint arXiv:2203.12204*, 2022.
- [20] J. G. Peter Byfield, Tuatini Godard, “Tools for tissue image stain normalisation and augmentation in python 3,” 2021.
- [21] A. Vahadane, T. Peng, A. Sethi, S. Albarqouni, L. Wang, M. Baust, K. Steiger, A. M. Schlitter, I. Esposito, and N. Navab, “Structure-preserving color normalization and sparse stain separation for histological images,” *IEEE transactions on medical imaging*, vol.35, no.8, pp.1962–1971, 2016.



شکل ۱۶: مقایسه نتایج دو مدل روی تصاویر ۱۰۲۴ در ۱۰۲۴



شکل ۱۷: مقایسه نتایج دو مدل روی تصاویر ۳۰۷۲ در ۲۰۷۲