

به نام خدا



دانشگاه تهران



دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین ششم

نام دستیار طراح	سعید رحیمی	پرسش ۱
رایانامه	Saeedrhimi2@gmail.com	
نام دستیار طراح	محمد کرمی	پرسش ۲
رایانامه	karami.m7906@gmail.com	
مهلت ارسال پاسخ	۱۴۰۴/۰۳/۳۱	

پرسش ۱. یادگیری بدون نظارت و انتقال دامنه با استفاده از GAN.....	۳
۱-۱ مقدمه.....	۳
۲-۱ بخش نظری (۳۰ نمره).....	۳
۳-۱ بخش عملی (۷۰ نمره).....	۴
پیش پردازش داده ها (۵ نمره).....	۴
آزمون مدل پایه و مشاهده Domain Gap (۵ نمره).....	۵
پیاده سازی معماری مدل (۲۰ نمره).....	۵
پیاده سازی توابع هزینه (۱۰ نمره).....	۶
آموزش مدل (۲۰ نمره).....	۶
نکته مهم در پیاده سازی Classifier.....	۷
نمایش نتایج (۱۰ نمره).....	۷
پرسش ۲. بازسازی تصاویر پولیپ آندوسکوپی با EndoVAE.....	۸
۱-۲. پیش پردازش داده ها (۱۰ نمره).....	۸
۲-۲. طراحی معماری EndoVAE (۲۰ نمره).....	۸
۳-۲. تعریف توابع هزینه (۱۰ نمره).....	۹
۴-۲. روند آموزش مدل (۲۵ نمره).....	۹
۵-۲. تولید و بازسازی کیفی (۱۵ نمره).....	۱۰
۶-۲. ارزیابی عددی (۱۰ نمره).....	۱۰
۷-۲. تحلیل نتایج و بحث نهایی (۱۰ نمره).....	۱۱
۸-۲. بخش امتیازی (۵ نمره).....	۱۱

قبل از پاسخ دادن به پرسش‌ها، موارد زیر را با دقت مطالعه نمایید:

- از پاسخ‌های خود یک گزارش در قالبی که در صفحه‌ی درس در سامانه‌ی Elearn با نام **REPORTS_TEMPLATE.docx** قرار داده شده تهیه نمایید.
- پیشنهاد می‌شود تمرین‌ها را در قالب گروه‌های دو نفره انجام دهید. (بیش از دو نفر مجاز نیست و تحویل تک نفره نیز نمره‌ی اضافی ندارد) توجه نمایید الزامی در یکسان ماندن اعضای گروه تا انتهای ترم وجود ندارد. (یعنی، می‌توانید تمرین اول را با شخص A و تمرین دوم را با شخص B و ... انجام دهید)
- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است؛** بنابراین، لطفا تمامی نکات و فرض‌هایی را که در پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید در گزارش ذکر کنید.
- در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
- **تحلیل نتایج الزامی می‌باشد، حتی اگر در صورت پرسش اشاره‌ای به آن نشده باشد.**
- **دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند؛** بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- **کدها حتما باید در قالب نوت‌بوک با پسوند ipynb تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتما در این فایل ارسالی شما ذخیره شده باشد.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده‌اید، این نمودار باید هم در گزارش هم در نوت‌بوک کدها وجود داشته باشد.
- **در صورت مشاهده‌ی تقلب نمره تمامی افراد شرکت‌کننده در آن، 100- لحاظ می‌شود.**
- تنها زبان برنامه نویسی مجاز **Python** است.
- **استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست.** در صورتی که دو گروه از یک منبع مشترک استفاده کنند و کدهای مشابه تحویل دهند، تقلب محسوب می‌شود.
- نحوه محاسبه تاخیر به این شکل است: پس از پایان رسیدن مهلت ارسال گزارش، حداکثر تا یک هفته امکان ارسال با تاخیر وجود دارد، پس از این یک هفته نمره آن تکلیف برای شما صفر خواهد شد.

○ سه روز اول: بدون جریمه

○ روز چهارم: ۵ درصد

○ روز پنجم: ۱۰ درصد

○ روز ششم: ۱۵ درصد

○ روز هفتم: ۲۰ درصد

- حداکثر نمره‌ای که برای هر سوال می‌توان اخذ کرد ۱۰۰ بوده و اگر مجموع بارم یک سوال بیشتر از ۱۰۰ باشد، در صورت اخذ نمره بیشتر از ۱۰۰، اعمال نخواهد شد.

○ برای مثال: اگر نمره اخذ شده از سوال ۱ برابر ۱۰۵ و نمره سوال ۲ برابر ۹۵ باشد، نمره نهایی

تمرین ۹۷.۵ خواهد بود و نه ۱۰۰.

- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه‌ی Elearn بارگذاری نمایید:

HW[Number]_[Lastname]_[StudentNumber]_[Lastname]_[StudentNumber].zip

(مثال: HW1_Ahmadi_810199101_Bagheri_810199102.zip)

- برای گروه‌های دو نفره، بارگذاری تمرین از جانب یکی از اعضا کافی است ولی پیشنهاد می‌شود هر دو نفر بارگذاری نمایند.

پرسش ۱. یادگیری بدون نظارت و انتقال دامنه با استفاده از GAN

۱-۱ مقدمه

یکی از چالش‌های اصلی در کاربرد عملی یادگیری عمیق، مسئله‌ای به نام شکاف دامنه (Domain Shift) است. این وضعیت زمانی رخ می‌دهد که مدلی که روی یک مجموعه داده (دامنه‌ی منبع) آموزش دیده، در مواجهه با داده‌هایی از دامنه‌ای دیگر (با تفاوت‌های ظاهری، شرایط نوری، سبک بصری و ...) دچار افت شدید عملکرد می‌شود. در بسیاری از کاربردهای واقعی، برچسب‌گذاری داده‌های جدید (مثلاً تصاویر واقعی) دشوار، پرهزینه یا حتی غیرممکن است؛ اما داده‌های مصنوعی یا ساده‌تر (مانند داده‌های تولیدشده یا شبیه‌سازی‌شده) به راحتی در دسترس هستند.

مقاله‌ی [Unsupervised Pixel-Level Domain Adaptation with GANs](#) دقیقاً به همین مسئله می‌پردازد. نویسندگان این مقاله راهکاری ارائه می‌دهند که با استفاده از شبکه‌های GAN، تصاویر دامنه‌ی منبع را به تصاویری با ظاهر دامنه‌ی هدف تبدیل می‌کند؛ به طوری که محتوای اصلی تصویر حفظ شود ولی ظاهر آن شبیه به داده‌های هدف باشد. این فرایند بدون استفاده از هیچ برچسبی در دامنه‌ی هدف انجام می‌شود.

مزیت کلیدی این روش، انتقال دانش از یک دامنه به دامنه‌ای دیگر در سطح تصویر (pixel-level) است؛ رویکردی که نه تنها باعث بهبود عملکرد مدل در دامنه‌ی هدف می‌شود، بلکه خروجی آن نیز قابل تفسیر و تحلیل است.

در این تمرین، شما با بازسازی این مقاله، هم با مفاهیم پیشرفته‌ای مانند domain adaptation و adversarial learning آشنا می‌شوید، و هم تجربه‌ی عملی طراحی مدل‌های ترکیبی و تحلیل عملکرد آن‌ها را کسب خواهید کرد.

۲-۱ بخش نظری (۳۰ نمره)

۱-۲-۱. چرا آموزش GAN ناپایدار است؟ سه عامل اصلی بی‌ثباتی را نام ببرید و هر کدام را توضیح

دهید. به نظر شما، چه مکانیزم‌هایی برای مقابله با این ناپایداری پیشنهاد شده‌اند؟ (۵ نمره)

۱-۲-۲. در یک GAN معمولی، Generator فقط نویز z را به تصویر تبدیل می‌کند. در این مقاله،

Generator به جای آن، از یک تصویر ورودی و نویز استفاده می‌کند. به نظر شما این تغییر چه

اثری بر یادگیری و کنترل خروجی دارد؟ (۵ نمره)

۱-۲-۳. مدل معرفی شده در این مقاله شامل سه مؤلفه اصلی است Discriminator ، Generator و

Classifier . به نظر شما، نقش هر کدام در یادگیری چه چیزی است؟ اگر یکی از آن‌ها حذف

شود، عملکرد کل مدل چه تغییری می‌کند؟ (۵ نمره)

۱-۲-۴. یکی از ویژگی‌های خاص مدل معرفی شده در این مقاله، استفاده از content similarity

loss (برای حفظ محتوا) است. توضیح دهید چرا حفظ محتوای تصویر منبع ضروری است، و

اگر این مکانیزم حذف شود، چه رفتاری از Generator انتظار دارید؟ (۵ نمره)

۱-۲-۵. مدل معرفی شده در این مقاله از یک طبقه‌بند مستقل برای آموزش همزمان روی تصویر

اصلی و تصویر تولیدی استفاده می‌کند. این کار چه مزیتی نسبت به آموزش فقط روی تصاویر

تولیدی دارد؟ تحلیل کنید چگونه این انتخاب به پایداری یادگیری کمک می‌کند. (۵ نمره)

۱-۲-۶. روش این مقاله برای دامنه‌هایی مانند MNIST و MNIST-M طراحی شده که تفاوت

آن‌ها در سبک (style) است. آیا می‌توان از همین روش برای دامنه‌هایی که تفاوت semantic

دارند (مثلاً اشیای متفاوت، زبان متفاوت، یا زاویه دید متفاوت) استفاده کرد؟ استدلال کنید.

(۵ نمره)

۱-۳ بخش عملی (۷۰ نمره)

پیش پردازش داده‌ها (۵ نمره)

در این تمرین از دو مجموعه داده MNIST و MNIST-M استفاده می‌کنیم که می‌توانید آن‌ها را از [این](#)

[لینک](#) دانلود کنید. مجموعه‌ی MNIST شامل تصاویر سیاه‌سفید از ارقام ۰ تا ۹ است و به‌عنوان دامنه‌ی

منبع (source domain) استفاده می‌شود. مجموعه‌ی MNIST-M نسخه‌ی تغییر یافته‌ای از همین تصاویر

است که با اضافه شدن پس‌زمینه‌های رنگی ایجاد شده و به‌عنوان دامنه‌ی هدف (target domain) در نظر

گرفته می‌شود. ترتیب تصاویر در هر دو مجموعه حفظ شده است، به این معنا که هر تصویر در MNIST

دقیقاً با همان اندیس در MNIST-M وجود دارد و تنها تفاوت آن‌ها در ظاهر (پس‌زمینه و رنگ) است، نه

در محتوای عددی.

در ابتدا، ۵ نمونه‌ی متناظر از هر دو مجموعه را با استفاده از اندیس یکسان انتخاب کرده و تصاویر

MNIST و MNIST-M را به‌صورت ردیفی و کنار هم نمایش دهید تا تفاوت ظاهری آن‌ها و حفظ محتوا

قابل مشاهده باشد.

سپس باید داده‌ها را مطابق با دستورالعمل مقاله آماده‌سازی کرده و در DataLoader های جداگانه قرار دهید. ابتدا تصاویر MNIST که به صورت تک کاناله (grayscale) هستند را به تصاویر سه کاناله (RGB) تبدیل کنید تا با ساختار داده‌های MNIST-M هماهنگ شوند. هر دو مجموعه باید به ابعاد مورد نیاز معماری مدل (مثلاً 32×32) resize شوند. پس از آن، تمام تصاویر باید به بازه‌ی $[-1, 1]$ نرمال شوند. در گام بعد، هر کدام از دو مجموعه داده را به دو بخش آموزش و آزمون با نسبت ۸۰٪ آموزش و ۲۰٪ آزمون تقسیم کنید. چون ترتیب تصاویر در هر دو مجموعه یکسان است، باید تقسیم‌بندی با استفاده از اندیس‌های مشترک انجام شود تا تناظر بین نمونه‌های MNIST و MNIST-M حفظ شود. در نهایت، برای هر مجموعه، دو DataLoader جداگانه تعریف کنید: یکی برای داده‌های آموزش و یکی برای داده‌های آزمون. در مجموع باید چهار DataLoader ساخته شود: آموزش و آزمون برای MNIST و آموزش و آزمون برای MNIST-M. جزئیات مربوط به پیش‌پردازش و ساخت batch ها را از متن مقاله استخراج و در پیاده‌سازی خود رعایت کنید.

آزمون مدل پایه و مشاهده Domain Gap (۵ نمره)

یک کلاسیفایر با معماری دقیق مقاله (کلاسیفایر MNIST-M)، فقط روی داده‌های MNIST آموزش دهید. سپس مدل آموزش‌دیده را بدون هیچ تغییر، روی داده‌های تست MNIST و کل داده‌های MNIST-M ارزیابی و دقت هر دو را گزارش کنید. اختلاف دقت را به اختصار تحلیل کنید. توجه: معماری باید مطابق مقاله باشد.

پیاده‌سازی معماری مدل (۲۰ نمره)

پس از آماده‌سازی داده‌ها، شما باید سه بخش اصلی مدل را مطابق توضیحات مقاله طراحی و پیاده‌سازی کنید. این سه بخش شامل یک **generator** برای انتقال سبک تصویر، یک **discriminator** برای تشخیص واقعی یا جعلی بودن تصویر، و یک **classifier** برای انجام وظیفه طبقه‌بندی است. ساختار هر سه مدل در مقاله توضیح داده شده و باید مطابق آن طراحی شوند. جزئیات معماری‌ها، تعداد لایه‌ها، نوع نرمال‌سازی، نوع فعال‌سازی، و مسیر داده در هر شبکه را از متن مقاله و شکل‌های ارائه‌شده استخراج کرده و در قالب کد پیاده‌سازی کنید.

پیاده سازی توابع هزینه (۱۰ نمره)

پس از طراحی و پیاده سازی معماری مدل ها، در این مرحله لازم است توابع هزینه (loss functions) مورد استفاده در آموزش مدل را مطابق مقاله پیاده سازی کنید. مقاله دو نوع loss اصلی معرفی می کند که هر کدام نقش متفاوتی در بهینه سازی بخش های مختلف مدل دارند. اولین loss، **adversarial loss** است که تعامل رقابتی بین Generator و Discriminator را هدایت می کند و به Generator یاد می دهد چگونه تصاویر تولیدی را از دیدگاه Discriminator واقعی جلوه دهد. دوم، **task loss** یا همان classification loss است که عملکرد Classifier را روی تشخیص درست ارقام (در هر دو تصویر اصلی و تولیدی) ارزیابی می کند. شما باید این loss ها را به صورت جداگانه تعریف کرده و مطابق با توضیح مقاله، در مراحل آموزش Generator، Discriminator و Classifier استفاده کنید. ضرایب ترکیب این loss ها نیز باید به صورت هایپرپارامتر در نظر گرفته شود. نحوه ترکیب دقیق این loss ها در جریان آموزش را از متن مقاله استخراج و پیاده سازی کنید.

آموزش مدل (۲۰ نمره)

در این بخش باید مدل را مطابق با ساختار آموزشی مقاله آموزش دهید. Generator تنها با تصاویر دامنه ی منبع (MNIST) آموزش می بیند: تصویر MNIST به همراه نویز به آن داده می شود تا تصویری با ظاهر دامنه ی هدف (MNIST-M) تولید کند که محتوای عددی را حفظ کرده باشد. Classifier نیز فقط روی تصاویر دامنه ی منبع آموزش داده می شود، هم روی تصاویر اصلی MNIST و هم تصاویر فیک که از آن ها تولید شده اند. در هیچ مرحله ای از آموزش، نباید از برجسب های MNIST-M (دامنه هدف) استفاده شود. Discriminator به طور جداگانه آموزش می بیند و باید تفاوت بین تصاویر واقعی MNIST-M و تصاویر تولید شده را یاد بگیرد.

در هر epoch، مقادیر loss مربوط به Generator، Discriminator و Classifier و همچنین دقت Classifier باید روی داده های train و test اندازه گیری و ذخیره شوند، چرا که در مرحله بعدی باید آن ها را نمایش و تحلیل کنید. جزئیات ترتیب به روزرسانی، نحوه ی ترکیب loss ها و داده های مورد استفاده را مطابق مقاله پیاده سازی کنید و مدل را آموزش دهید. (برای راحتی آموزش میتوانید مدل را بعد از هر اپیک یا تعداد مشخصی اپیک به همراه داده های دیگر ذخیره کنید تا در صورت بروز مشکل از همانجا عملیات آموزش را ادامه دهید).

نکته مهم در پیاده‌سازی Classifier:

دقت کنید که رفتار طبقه‌بند در فاز آموزش و تست متفاوت است و باید این تفاوت را در کد به‌درستی اعمال کنید. در فاز آموزش، Classifier فقط با داده‌های دامنه منبع (MNIST) آموزش می‌بیند؛ آن هم صرفاً روی تصاویر واقعی MNIST و تصاویر فیک که از آن‌ها توسط Generator ساخته شده‌اند، و فقط از برچسب‌های MNIST استفاده می‌شود. در این فاز هیچ‌گونه استفاده‌ای از داده‌های MNIST-M یا برچسب‌های آن مجاز نیست.

در مقابل، در فاز تست، عملکرد Classifier باید به‌صورت جداگانه روی داده‌های test MNIST (منبع) و MNIST-M (هدف) ارزیابی شود. در این مرحله استفاده از برچسب‌های MNIST-M فقط برای ارزیابی دقت مجاز است، نه برای یادگیری. این تفکیک بسیار مهم است؛ هرگونه استفاده از برچسب‌های target در فاز آموزش، خلاف فرض unsupervised domain adaptation بوده و نتیجه‌ی مدل را بی‌اعتبار می‌کند.

نمایش نتایج (۱۰ نمره)

در این بخش، باید نتایج loss و accuracy را که در طول آموزش ذخیره کرده‌اید، به‌صورت نمودار نمایش دهید.

در کنار نمودارها، باید چند نمونه تصویر برای ارزیابی کیفی مدل نمایش دهید. پنج تصویر از مجموعه test در MNIST انتخاب کنید و آن‌ها را به Generator بدهید تا تصویر فیک تولید شود. سپس برای هر نمونه، تصاویر زیر را به‌صورت ردیفی و متناظر کنار هم نمایش دهید: تصویر اصلی MNIST، تصویر تولیدشده توسط Generator، تصویر واقعی متناظر از MNIST-M. در کنار هر تصویر، برچسب واقعی و پیش‌بینی‌شده توسط Classifier را نیز بنویسید. ترتیب نمایش باید حفظ شود تا بتوان بررسی کرد آیا Generator ظاهر تصویر را درست تغییر داده، محتوای عددی را حفظ کرده، و Classifier عدد را درست تشخیص داده است یا نه.

پرسش ۲. بازسازی تصاویر پولیپ آندوسکوپی با EndoVAE

در این پرسش باید معماری مقاله [EndoVAE](#) را بازسازی کرده و با استفاده از دیتاست [Kvasir](#) ارزیابی نمایید. توجه کنید که به جای دیتاست اصلی مقاله (KID)، از دیتاست Kvasir استفاده می‌کنید

۱-۲. پیش‌پردازش داده‌ها (۱۰ نمره)

دیتاست Kvasir را دانلود و استخراج کنید و سپس تصاویر سالم را از پوشه‌های زیر انتخاب کنید سعی کنید از همه تصاویر سالم استفاده کنید و استفاده از دیتا آگمنتیشن برای افزایش تصاویر سالم میتواند کمک کننده باشد.

- normal-z-line
- normal-pylorus
- normal-cecum

تصاویر پولیپ را از پوشه polyps قسمت Kvasir-SEG بردارید.

سپس مطابق مقاله ، همه تصاویر را بارگذاری و به فضای رنگی RGB تبدیل کنید و همچنین اندازه هر تصویر را به صورت bilinear به 96×96 تغییر دهید. و شدت پیکسل‌ها را به بازه $[0,1]$ نرمال کنید . همچنین اگر پیش پردازش دیگری مانند کراپ کردن و .. مناسب است با ذکر دلیل در گزارش انجام دهید. مطالعه بخش A از بخش چهارم مقاله در این بخش میتواند کمک کننده باشد.

در ادامه خروجی نهایی را در دو پوشه جدید ذخیره کنید:

processed/normal (تمام تصاویر سالم، ابعاد $3 \times 96 \times 96$)

processed/polyp (تمام تصاویر پولیپ، ابعاد $3 \times 96 \times 96$)

در گزارش دست‌کم یک نمونه از تصویر قبل و بعد از این پیش‌پردازش را نشان دهید.

۲-۲. طراحی معماری EndoVAE (۲۰ نمره)

مطابق بخش‌های سه مقاله و شکل یک پیاده سازی را انجام دهید

Encoder: یک شبکه کانولوشنی شامل شش لایه Conv2D+ReLU که خروجی‌اش را flatten

کرده و با دو مسیر FC به بردارهای μ و $\log\sigma$ با طول $\text{latent_dim} = 6$ می‌رساند.

تابع `eparameterize(mu, logvar)` طبق توضیحات مقاله

Decoder: ابتدا بردار z را با یک لایه FC به تانسور $6 \times 6 \times 256$ گسترش داده و سپس با هفت لایه `ConvTranspose2d+ReLU` (و `Sigmoid` در لایه آخر) خروجی را به تصویر $96 \times 96 \times 3$ برگرداند

مطمئن شوید ابعاد میانی لایه‌ها دقیقاً منطبق با شکل یک مقاله باشد همچنین در گزارش برداشت خود از هر کدام از بخش‌های مهم معماری را توضیح دهید.

۳-۲. تعریف توابع هزینه (۱۰ نمره)

مطابق بخش سه مقاله توابع هزینه را تعریف کنید همینطور در گزارش برداشت خود از هر کدام را بنویسید.

Reconstruction Loss: استفاده از `Binary Cross-Entropy (BCE)` بین x و \hat{x} می‌توانید `MSE` را نیز تست کنید و در گزارش با دلیل یا مقایسه بنویسید از کدام استفاده کرده اید
KL Divergence: فرمول استاندارد VAE از همان بخش مقاله.

جمع نهایی `total_loss` :

$$\text{Reconstruction} + 1.0 \times \text{KL total_loss} = \text{Reconstruction} + 1.0 \times \text{KL}$$

۴-۲. روند آموزش مدل (۲۵ نمره)

هایپرپارامترها را طبق بخش A.4 مقاله تنظیم کنید.

داده‌های «سالم» را در یک `Dataset` و `DataLoader` با `shuffle=True` قرار دهید.

روند آموزشی را مطابق مقاله انجام دهید:

۱. مدل در حالت `train`

۲. فوروارد از Encoder \rightarrow reparameterize \rightarrow Decoder

۳. محاسبه Reconstruction Loss و KL Loss و جمع زدن آنها

۴. optimizer.step و backprop

۵. هر چند epoch یکبار میانگین هر Loss را یادداشت کنید.

گزارستان از این بخش باید به خوبی نشان دهد روند آموزش را به درستی و طبق مقاله انجام داده اید.

۲-۵. تولید و بازسازی کیفی (۱۵ نمره)

۲-۵-۱. تولید

مدل آموزش دیده را در حالت eval بارگذاری کنید و مطابق بند 4 مقاله:

نه عدد بردار $z \sim N(0, I)$ تصادفی بسازید. هر یک را به Decoder بدهید و مجموعه ۹ تصویر تولیدشده را بگیرید سپس یک گرید 3×3 از این ۹ تصویر در گزارش نمایش دهید.

۲-۵-۲. بازسازی

یک Batch از ۵ تصویر پولیپ از پوشه processed/polyp انتخاب کنید و سپس مطابق توضیحات مقاله تصاویر را به Encoder بدهید تا $(\mu, \log \phi)$ را بگیرد، سپس با Decoder بازسازی کنید. سپس یک گرید 2×5 (در هر ردیف: تصویر واقعی پولیپ و تصویر بازسازی شده) در گزارش نمایش دهید.

۲-۶. ارزیابی عددی (۱۰ نمره)

برای ۵۰ تصویر تصادفی از پوشه processed/polyp:

هر تصویر را بازسازی کنید.

PSNR و SSIM بین تصویر واقعی و تصویر بازسازی شده را محاسبه نمایید.

برای آن ۵۰ نمونه، جدولی تهیه کنید با دو ستون: «میانگین» و «انحراف معیار» برای PSNR و SSIM.

توجه شود این معیار ها در مقاله نیست و برای تحلیل کمی بازسازی خود از آنها استفاده میکنیم.

۷-۲. تحلیل نتایج و بحث نهایی (۱۰ نمره)

تحلیل کیفی:

- تصاویر تولیدشده تا چه حد واقع‌گرایانه‌اند؟
- بازسازی پولیپ چه جزئیاتی را حفظ کرده و چه بخش‌هایی محو شده‌اند؟

تحلیل کمی:

- مقادیر PSNR و SSIM چه نکاتی درباره کیفیت بازسازی نشان می‌دهند؟
- اگر مقادیر PSNR و SSIM پایین هستند، احتمالاً مشکل در کدام بخش (معماری، هایپرپارامتر یا پیش‌پردازش) است؟

به دلیل تفاوت در دیتاست هدف از این پیاده سازی این نیست که نتایجی برابر مقاله داشته باشیم بلکه درک شما از معماری مقاله و تحلیل نتایج معیار نمره دهی میباشد.

۸-۲. بخش امتیازی (۵ نمره)

یک کلاس‌بند ساده CNN تعریف کنید تا بین «پولیپ واقعی» و «پولیپ بازسازی‌شده» تمایز قائل شود.

۲۰۰ تصویر واقعی پولیپ و ۲۰۰ تصویر بازسازی‌شده را در دو دسته train و test قرار دهید.

دقت (Accuracy) و AUC آن را محاسبه کنید و در گزارش بنویسید.

تحلیل کنید آیا تصاویر بازسازی‌شده برای آموزش کلاس‌بند تقریباً شبیه واقعی‌ها بوده‌اند یا خیر.