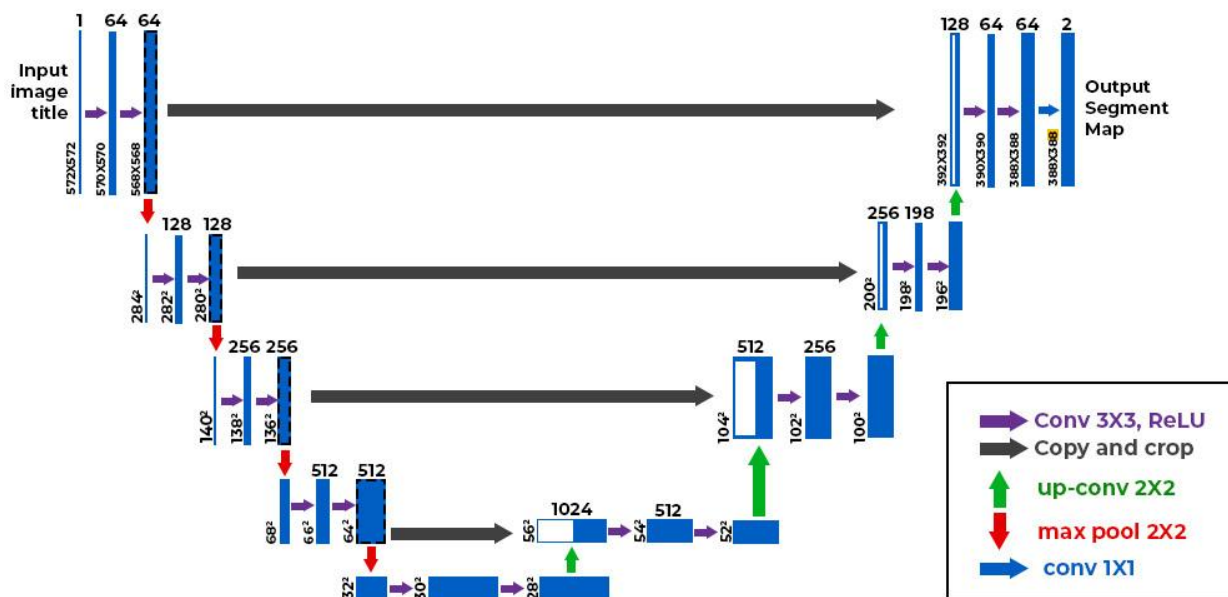


بینایی کامپیوتر – تمرین سری ششم

سوال هفتم – مهرشاد فلاح اسطخزیر

۴۰۱۵۲۱۴۶۲

منطق کد:



ابتدا دو بلاک encoder و decoder طراحی کردم. بر این اساس که در بلاک encoder دو فیلتر کانولوشن ۳ در ۳ زده و یک MaxPool با stride برابر با ۲ تا ابعاد نصف شود و downsampling رخ دهد. خروجی بلاک encoder به ترتیب برای skip features (که بعدا با concatenate شدن در مرحله decode بلاکها را upSample می کنند) و p که نتیجه maxPool است. در بلاک decoder هم که کار بازیابی اندازه اصلی عکس است. یک فیلتر ترانهاده کانولوشن زده و بعد از concatenate کردن با ویژگی های سطح پایین در مرحله encode دو فیلتر کانولوشن ۳ در ۳ می زنم. به همین ترتیب و با استفاده از این دو بلاک مدل U-net را می سازم با این تفاوت نسبت به عکس که از padding زیرو استفاده می کنم و مثل حالت عادی از valid استفاده نمی کنم (که همان بدون padding است). ۴ مرحله از بلاک encoder استفاده می کنم. در bottleneck هم دو مرتبه از فیلتر کانولوشن استفاده می کنم و خروجی گلوگاه (bottleneck) را به ورودی بلاک decoder می دهم. در نهایت برای خروجی از یک فیلتر کانولوشن با تابع فعال سازی softmax استفاده

می‌کنم. در حالت عادی باید ابعاد خروجی هم مشخص باشد. در این مثال ۲۱ کلاس داریم به همین دلیل به صورت static عدد ۲۱ را برای تعداد کلاس‌ها دادم. در نهایت هم مدل را ساخته.

در طراحی مدل از این [لینک](#) استفاده شد.

توضیحات مربوط به تابع ضرر [Dice](#) و [Focal](#) هم در لینک‌ها آمده است. همچنین راجع به `jaccard_coef` که یک امتیاز شباهت هست هم در لینک توضیحات موجود است. به طور کلی `jaccard_coef` بین ۰ تا ۱ است و هر چه به یک نزدیک‌تر باشد دو مجموعه مورد بررسی شباهت بیشتری دارند. `Jaccard_coef` و `IoU` فرمول‌های مشابهی دارند.

تحلیل نتایج:

مدل اصلی خودمان درصد دقت ۶۲.۴ درصدی داشت. مدل تعریف شده کتابخانه و با فریز کردن لایه‌های انکودر نزدیک به ۲ میلیون پارامتر قابل آموزش کمتر داشت و دقت ۷۰.۱ درصدی داشت و بدون فریز کردن لایه‌های انکودر دقت نزدیک به ۷۷.۴ درصدی. این نتایج نشان می‌دهد جزئیات پیاده‌سازی شده در کتابخانه نتیجه را بهتر می‌کند و همچنین این که اگر فریز نشود درصد بالاتر می‌رود.