به نام خدا

دانشگاه تهران

دانشکدگان علوم و فناوری های میان رشته ای

Neural Network

تمرین سوم

نام و نام خانوادگی	امید مرادی
شماره دانشجویی	۸۳۰۴۰۲۰۷۱
تاریخ ارسال گزارش	۱۹ دی

سوال اول

Vision Transformers (ViTs)

Vision Transformers (ViTs) معماری‌های شبکه عصبی هستند که برای پردازش تصاویر با استفاده از اصول مدل‌های ترانسفورماتور طراحی شده‌اند که در اصل برای پردازش زبان طبیعی (NLP) توسعه یافته‌اند.

ترانسفورماتورهای بینایی کاربردهای گسترده ای در کارهای رایج تشخیص تصویر مانند تشخیص اشیا، تقسیم بندی تصویر، طبقه بندی تصویر و تشخیص عمل دارند. علاوه بر این، ViTها در مدل‌سازی تولیدی و وظایف چند مدلی، از جمله زمینه‌سازی بصری، پاسخ‌گویی به سوال بصری، و استدلال بصری استفاده می‌شوند.

در ViTs، تصاویر به صورت توالی نمایش داده می‌شوند و برچسب‌های کلاس برای تصویر پیش‌بینی می‌شوند که به مدل‌ها اجازه می‌دهد تا ساختار تصویر را به طور مستقل یاد بگیرند. تصاویر ورودی به‌عنوان دنباله‌ای از وصله‌ها در نظر گرفته می‌شوند که در آن هر وصله با به هم پیوستن کانال‌های همه پیکسل‌ها در یک وصله و سپس نمایش خطی آن به بعد ورودی مورد نظر، به یک بردار واحد تبدیل می‌شود.

ساختار ViT:

1. **Image Tokenization**
2. **Linear Embedding**
3. **Positional Embeddings**
4. **Transformer Encoder**
5. **Classification Token**
6. **Output Layer**

ویژگی های اصلی:

ورودی مبتنی بر پچ:

ViT ها وصله های تصویر را به جای کل تصویر پردازش می کنند و به مدل این امکان را می دهند که اندازه های متغیر تصویر را مدیریت کند و به طور مستقل روی مناطق فضایی تمرکز کند.

مکانیسم توجه:

ماژول MHSA به مدل اجازه می دهد تا وابستگی های دوربرد را بیاموزد و بر روی قسمت های خاصی از تصویر که مربوط به یک کار است تمرکز کند.

مقیاس پذیری:

ViTها با مجموعه داده‌ها و مدل‌های بزرگ (به عنوان مثال، ترانسفورماتورهای عمیق‌تر و گسترده‌تر) به خوبی مقیاس می‌شوند، زیرا مکانیسم توجه به طور مؤثر اطلاعات جهانی را مدیریت می‌کند.

پیش آموزش و تنظیم دقیق:

ViTها زمانی بهترین عملکرد را دارند که روی مجموعه داده های مقیاس بزرگ (مانند ImageNet-21k، JFT-300M) از قبل آموزش داده شوند و سپس روی مجموعه داده های کوچکتر و مختص کار تنظیم شوند.

تعصبات استقرایی کمتر:

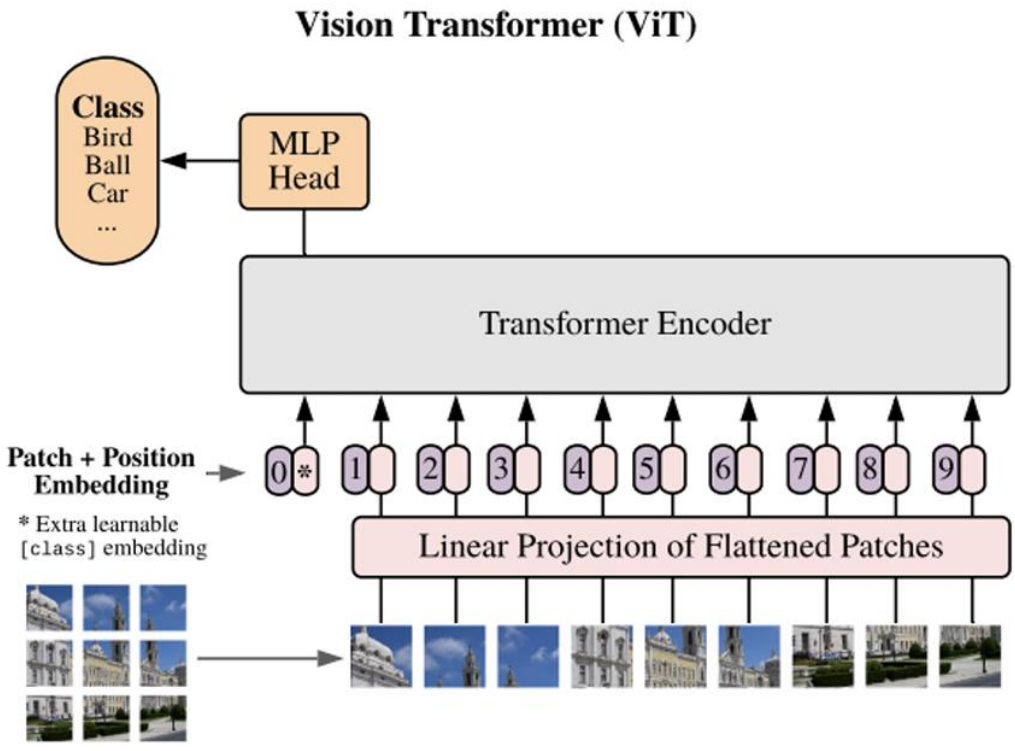
برخلاف شبکه‌های عصبی کانولوشنال (CNN)، ViTها به میدان‌های دریافتی محلی یا اشتراک وزن متکی نیستند. این باعث می‌شود آنها انعطاف‌پذیرتر باشند، اما در عین حال تشنه اطلاعات هستند.

عملکرد بالا:

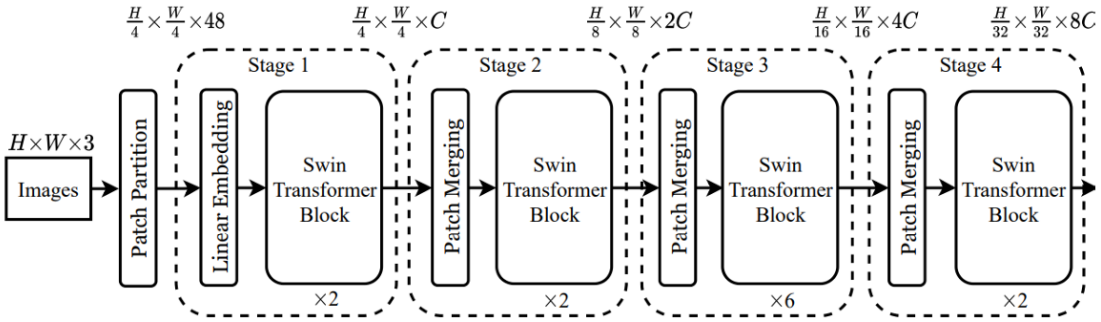
هنگامی که ViT ها به اندازه کافی از قبل آموزش دیده باشند، در بسیاری از وظایف بینایی کامپیوتری، از جمله طبقه بندی، تشخیص اشیا، و تقسیم بندی، به عملکردی پیشرفته دست می یابند.

انعطاف پذیری در بین روش ها:

معماری ترانسفورماتور را می توان با سایر مدالیته ها مانند متن تطبیق داد و ViTs را به یک مدل متحد کننده برای یادگیری چندوجهی تبدیل کرد.



Swin Transformers



این نمودار معماری ترانسفورماتور Swin را نشان می دهد. با تصویری به ابعاد H×W×3 شروع می شود که به تکه های ۴×۴ غیر همپوشانی (Patch Partition) تقسیم می شود. سپس این تکه‌ها به صورت خطی در بردارهای مشخصه بعد C جاسازی می‌شوند و در نتیجه نمایشی میانی با اندازه H/4×W/4×C ایجاد می‌شود.

این نمودار معماری ترانسفورماتور Swin را نشان می دهد. با تصویری به ابعاد H×W×3 شروع می شود که به تکه های ۴×۴ غیر همپوشانی (Patch Partition) تقسیم می شود. سپس این تکه‌ها به صورت خطی در بردارهای مشخصه بعد C جاسازی می‌شوند و در نتیجه نمایشی میانی با اندازه H/4×W/4×C ایجاد می‌شود.

معماری به چهار مرحله تقسیم می شود:

۱. مرحله ۱: بلوک های ترانسفورماتور Swin Patch های تعبیه شده را بدون تغییر اندازه فضایی آنها (H/4×W/4) پردازش می کنند، اما نمایش ویژگی را افزایش می دهند.
۲. مرحله ۲: ادغام وصله ابعاد فضایی را به H/8×W/8 کاهش می‌دهد، ابعاد ویژگی را دو برابر می‌کند و به ۲ Cمی‌رساند و به دنبال آن بلوک‌های ترانسفورماتور Swin بیشتر می‌شود.
۳. مرحله ۳: یکی دیگر از مراحل ادغام وصله، ابعاد فضایی را به H/16×W/16 کاهش می‌دهد، ابعاد ویژگی را به ۴ Cافزایش می‌دهد و بلوک‌ها این نمایش را پردازش می‌کنند.
۴. مرحله ۴: ادغام نهایی ابعاد فضایی را به H/32×W/32 با ابعاد ویژگی ۸ Cکاهش می‌دهد و به دنبال آن بلوک‌های ترانسفورماتور.Swin

معماری به تدریج وضوح فضایی را کاهش می دهد در حالی که ابعاد ویژگی را برای استخراج کارآمد ویژگی افزایش می دهد.

ویژگی های شبکه:

۱. نمایش سلسله مراتبی: شبکه تصاویر را در چهار مرحله پردازش می کند و به تدریج ابعاد فضایی را کاهش می دهد (از H/4×W/4 به H/32×W/32) در حالی که ابعاد ویژگی را

افزایش می دهد) از C به ۸ .(Cاین رویکرد سلسله مراتبی به طور موثر ویژگی های سطح پایین و بالا را به تصویر می کشد.

۲. ورودی مبتنی بر وصله: تصویر ورودی به وصله های غیر همپوشانی (۴×۴) تقسیم می شود که به صورت خطی در بردارهای ویژگی جاسازی شده اند. این به ترانسفورماتور اجازه می دهد تا تصاویر را به صورت توالی، مشابه مدل های پردازش زبان طبیعی، مدیریت کند.

۳. توجه مبتنی بر پنجره: هر بلوک ترانسفورماتور Swin توجه خود را در پنجره های محلی محاسبه می کند و پیچیدگی محاسباتی را در مقایسه با توجه جهانی کاهش می دهد. این باعث می شود شبکه برای تصاویر با وضوح بالا کارآمد باشد.

۴. مکانیسم پنجره جابجا شده: با جابجایی پنجره ها در لایه های متناوب، شبکه تعاملات بین پنجره ای را فعال می کند و توانایی آن را برای مدل سازی وابستگی های جهانی افزایش می دهد.

۵. ادغام وصله: در هر انتقال مرحله، وصله‌های مجاور ادغام می‌شوند و اندازه فضایی را کاهش می‌دهند و بعد ویژگی را افزایش می‌دهند. این پردازش چند مقیاسی شبکه‌های عصبی کانولوشنال (CNN) را تقلید می‌کند.

۶. مقیاس پذیری: شبکه را می توان با تنظیم تعداد بلوک ها در هر مرحله، بعد ویژگی C و اندازه وصله های ورودی، مقیاس پذیر کرد و آن را با وظایف مختلف و بودجه های محاسباتی سازگار کرد.

۷. تطبیق پذیری: ساختار سلسله مراتبی و مکانیسم های توجه آن را برای کارهای بینایی متنوع، از جمله طبقه بندی تصویر، تشخیص اشیا و تقسیم بندی معنایی مناسب می کند.

۸. کارایی: با محدود کردن توجه به پنجره های کوچکتر و کاهش تدریجی اندازه فضایی، ترانسفورماتور Swin به پیچیدگی خطی نسبت به اندازه تصویر دست می یابد و از نظر محاسباتی کارآمد می شود.

EfficientNet

این تصویر معماری EfficientNet را نشان می‌دهد که یک تصویر ورودی با اندازه ۳×۲۲۴×۲۲۴ را در یک نقشه ویژگی پردازش می‌کند. از اجزای زیر تشکیل شده است:

۱. پیچیدگی اولیه (Conv 3x3): یک لایه کانولوشن استاندارد با هسته ۳×۳ تصویر ورودی را پردازش می کند و ویژگی های فضایی اساسی را استخراج می کند.

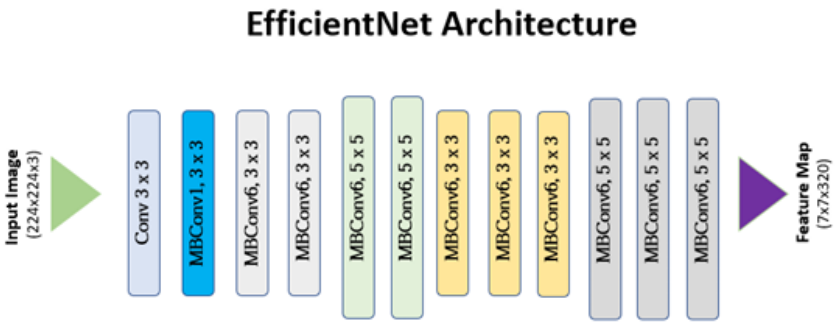
۲. بلوک‌های MBConv: اینها لایه‌های پیچیدگی گلوگاه معکوس موبایل (MBConv) با پیچیدگی‌های قابل تفکیک عمیق هستند که برای کاهش پیچیدگی محاسباتی طراحی شده‌اند.

۱. MBConv1 با هسته ۳×۳ عملیات سبک وزن را در اوایل شبکه معرفی می کند.
۲. MBConv6 با هسته‌های ۳×۳ یا ۵×۵ عمق می‌افزاید و فضای ویژگی را با استفاده از ضریب گسترش ۶ برابری برای ابعاد ویژگی میانی افزایش می‌دهد.

۳. استخراج ویژگی های پیشرونده: بلوک های MBConv به طور متناوب بین هسته های ۳×۳ و ۵×۵ برای گرفتن اطلاعات متنی ریز و بزرگتر قرار می گیرند. وضوح فضایی به تدریج کاهش می یابد و کانال های ویژگی با عمیق شدن شبکه افزایش می یابد.

۴. فیچرمپ نهایی: خروجی یک نقشه ویژگی به ابعاد ۷×۷×۳۲۰ است که ویژگی های سطح بالای تصویر ورودی را خلاصه می کند.

این طراحی با مقیاس‌بندی سیستماتیک عمق، عرض و وضوح، کارایی و عملکرد مدل را متعادل می‌کند و آن را برای کارهای طبقه‌بندی بسیار موثر می‌سازد.



ویژگی ها

EfficientNet برای عملکرد بالا با حداقل هزینه محاسباتی طراحی شده است. از مقیاس بندی ترکیبی برای تعادل سیستماتیک عمق (تعداد لایه ها)، عرض (تعداد کانال ها) و

وضوح (اندازه تصویر ورودی) برای مقیاس بندی کارآمد مدل استفاده می کند. این معماری دارای بلوک‌های MBConv است که از پیچیدگی‌های قابل تفکیک عمیق برای کاهش محاسبات و حفظ دقت استفاده می‌کنند. این هسته های ۳×۳ و ۵×۵ را برای استخراج ویژگی های متنوع ترکیب می کند و به تدریج ابعاد فضایی را کاهش می دهد در حالی که کانال های ویژگی را برای نمایش سلسله مراتبی ویژگی افزایش می دهد. EfficientNet با پارامترهای کمتر و FLOP کمتر در مقایسه با سایر شبکه ها به نتایج پیشرفته ای در طبقه بندی تصاویر دست می یابد.

سوال دوم

روش ارائه‌شده در مقاله برای تخمین عدم قطعیت تصمیم‌گیری شبکه‌های عصبی، از تزریق دراپ‌آوت در زمان تست به‌عنوان روشی برای کمّی‌سازی پس‌پرداز عدم قطعیت استفاده می‌کند. برخلاف دراپ‌آوت تعبیه‌شده سنتی که لایه‌های دراپ‌آوت در هر دو مرحله آموزش و تست فعال هستند، این روش لایه‌های دراپ‌آوت را به شبکه‌ای که قبلاً آموزش دیده است اضافه کرده و آن‌ها را تنها در مرحله تست فعال می‌کند. این امکان را فراهم می‌آورد تا بدون نیاز به آموزش شبکه با استفاده از دراپ‌آوت، عدم قطعیت تخمین زده شود و از بار محاسباتی ناشی از آموزش مجدد جلوگیری شود.

در این روش، از مونت‌کارلو دراپ‌آوت در زمان تست استفاده می‌شود که طی آن چندین پیش‌گذر تصادفی در شبکه انجام می‌شود. این فرآیند به تخمین توزیع پیش‌بینی‌های شبکه و عدم قطعیت مربوطه کمک می‌کند. میانگین و واریانس خروجی‌های تصادفی، به ترتیب پیش‌بینی و عدم قطعیت را نشان می‌دهند. برای بهبود کیفیت این تخمین‌ها، نرخ دراپ‌آوت بهینه از طریق کمینه‌سازی خطای لگاریتمی منفی (NLL) روی یک مجموعه اعتبارسنجی تنظیم می‌شود. علاوه بر این، یک ضریب مقیاس‌بندی معرفی می‌شود تا عدم قطعیت‌ها بر اساس رابطه بین خطای پیش‌بینی و عدم قطعیت تخمینی تنظیم شوند. این مقیاس‌بندی اطمینان حاصل می‌کند که تخمین‌های عدم قطعیت قوی و کالیبره‌شده باقی می‌مانند.

این روش از این جهت حائز اهمیت است که می‌توان آن را بر روی هر شبکه‌ای که قبلاً آموزش دیده اعمال کرد و نیاز به دراپ‌آوت در مرحله آموزش را از بین برد و بدین ترتیب تلاش محاسباتی را به طور قابل توجهی کاهش داد. نتایج تجربی در مسائل رگرسیون و وظایف چالش‌برانگیزی مانند شمارش جمعیت نشان می‌دهد که این روش به‌طور مؤثری عدم قطعیت را کمّی‌سازی کرده و در عین حال دقت پیش‌بینی رقابتی را حفظ می‌کند. با مقیاس‌بندی مناسب مقادیر عدم قطعیت، این روش جایگزینی عملی و کارآمد برای دراپ‌آوت تعبیه‌شده در زمینه کمّی‌سازی عدم قطعیت ارائه می‌دهد.

سوال سوم

کافیست با یک مثال نشان دهیم این رابطه کار میکند:

با فرض یک تصویر ۲۴۰ در ۲۴۰ با گام ۲ و پدینگ ۱ و کرنل ۳ در ۳ داریم:

H’ and W’ = [(240 – 3 + 2 * 1) / 2] + 1 = ۱۲۰

سوال چهارم

الف) لایه‌های پیچشی (Convolutional Layers) و لایه‌های تماما متصل (Fully Connected Layers)در شبکه‌های عصبی پیچشی (CNN) تفاوت‌های کلیدی دارند. لایه‌های پیچشی تنها با یک بخش کوچک از ورودی ارتباط دارند و وزن‌ها به صورت اشتراکی بین تمام بخش‌های تصویر اعمال می‌شوند، در حالی که در لایه‌های تماما متصل، هر نورون به تمام نورون‌های لایه قبلی متصل است که باعث افزایش تعداد پارامترها می‌شود. لایه‌های پیچشی اطلاعات مکانی داده‌ها را حفظ کرده و برای استخراج ویژگی‌های محلی مانند لبه‌ها و بافت‌ها استفاده می‌شوند، در حالی که لایه‌های تماما متصل ارتباط مکانی را از بین می‌برند و بیشتر برای ترکیب ویژگی‌های استخراج‌شده و انجام تصمیم‌گیری نهایی، مانند طبقه‌بندی، مناسب‌اند. همچنین لایه‌های پیچشی به دلیل استفاده از کرنل‌های کوچک و وزن‌های اشتراکی تعداد پارامترهای کمتری دارند، اما در لایه‌های تماما متصل به دلیل اتصال کامل، پارامترهای بیشتری وجود دارد. لایه‌های پیچشی معمولا در مراحل اولیه شبکه برای یادگیری ویژگی‌های سطح پایین به کار می‌روند، در حالی که لایه‌های تماما متصل در انتهای شبکه برای یادگیری روابط پیچیده‌تر و تصمیم‌گیری نهایی استفاده می‌شوند.

ب)

برای استخراج فرمول تصحیح وزن‌ها در یک لایه کانولوشنی در شبکه‌های عصبی پیچشی (CNN)با استفاده از روش پس‌انتشار(Backpropagation) ، ابتدا باید گذر رو به جلو را بررسی کنیم. در این فرآیند، خروجی عملیات کانولوشن که با Zنشان داده می‌شود، به صورت زیر محاسبه می‌گردد:

$$Z=W*X+b$$

در این رابطه، Wوزن‌های فیلتر (کرنل)، Xنقشه ویژگی ورودی، و bبایاس هستند. پس از اعمال یک تابع فعال‌سازی غیرخطی g، خروجی نهایی به صورت A=g(Z)خواهد بود.

در فرآیند پس‌انتشار، هدف محاسبه گرادیان‌های تابع هزینه Lنسبت به وزن‌ها W، بایاس‌ها b، و ورودی Xاست. برای این کار، خطای منتقل‌شده از لایه‌های بعدی به لایه کانولوشنی برگردانده می‌شود. ترم خطا در لایه جاری که با $\delta^{(l)}$نشان داده می‌شود، گرادیان تابع هزینه نسبت به Zاست. با استفاده از قانون زنجیره‌ای، این ترم خطا به خطای لایه بعدی $\delta^{(l+1)}$و مشتق تابع فعال‌سازی $g'(Z)$مرتبط است.

گرادیان تابع هزینه نسبت به وزن‌های Wاز طریق انجام عملیات کانولوشن بین ورودی Xو ترم خطا $\delta^{(l)}$محاسبه می‌شود. این به صورت زیر بیان می‌گردد:

$$\partial L/\partial W=\delta(l)*X$$

در اینجا، *نشان‌دهنده عملیات همبستگی متقابل است که مشابه کانولوشن است اما بدون چرخاندن کرنل. این عملیات سهم هر منطقه ورودی در ترم خطا را تجمیع می‌کند و به گرادیان اجازه می‌دهد از طریق شبکه عبور کند.

گرادیان تابع هزینه نسبت به بایاس bساده‌تر بوده و از طریق جمع‌کردن ترم خطا $\delta^{(l)}$در تمام ابعاد مکانی محاسبه می‌شود:

$$\frac{\partial L}{\partial b} = \sum_{i,j} \delta_{i,j}^{(l)}$$

تصحیح وزن‌ها با استفاده از قانون به‌روزرسانی گرادیان نزولی انجام می‌شود. برای وزن‌ها، این قانون به صورت زیر است:

$$W\leftarrow W-\eta\partial L/\partial W$$

که در آن ηنرخ یادگیری است. با جای‌گذاری $\partial L/\partial W=\delta^{(l)}*X$، به فرمول نهایی تصحیح وزن‌ها می‌رسیم:

$$\Delta W=-\eta(\delta(l)*X)$$

این فرمول وزن‌های فیلتر کانولوشن را در حین آموزش برای به حداقل رساندن تابع هزینه تنظیم می‌کند و فرآیند پس‌انتشار در لایه کانولوشنی را تکمیل می‌کند.

ج)

فیلترها یا هسته‌ها در لایه‌های پیچشی (Convolutional Layers) با اعمال عملیات کانولوشن بر روی ورودی، به استخراج ویژگی‌های مختلف تصویر کمک می‌کنند. هر فیلتر به یک بخش کوچک از تصویر (receptive field)اعمال می‌شود و با جابجایی در سراسر تصویر، الگوهای خاصی مانند لبه‌ها، گوشه‌ها، بافت‌ها یا ویژگی‌های سطح بالاتر را شناسایی می‌کند. وزن‌های فیلترها در طول فرآیند آموزش به گونه‌ای به‌روزرسانی می‌شوند که فیلترها بتوانند ویژگی‌های خاص و مهم تصویر را یاد بگیرند. در لایه‌های اولیه شبکه، فیلترها معمولا ویژگی‌های ساده‌تر مانند لبه‌ها یا رنگ‌ها را شناسایی می‌کنند، در حالی که در لایه‌های عمیق‌تر، این فیلترها قادر به شناسایی ویژگی‌های پیچیده‌تر مانند اشکال، اجزا و ساختارهای کلی تصویر می‌شوند. بنابراین، فیلترها نقش کلیدی در تفکیک اطلاعات مفید از تصویر و آماده‌سازی آن برای پردازش و طبقه‌بندی ایفا می‌کنند.

د)

یکی از معایب و محدودیت‌های اصلی شبکه‌های عصبی پیچشی (CNN) در وظایف شناسایی اشیا (Object Detection)این است که این شبکه‌ها به طور ذاتی برای طبقه‌بندی طراحی شده‌اند و مکان اشیای مختلف را در تصویر مشخص نمی‌کنند. در مدل‌های اولیه مانندR-CNN ، مشکل اصلی سرعت پایین در پردازش تصاویر بود، زیرا این مدل برای هر ناحیه پیشنهادی (Region Proposal) یک‌بار شبکه عصبی را اجرا می‌کرد. این باعث افزایش زمان پردازش به ویژه برای تصاویر با تعداد زیادی ناحیه پیشنهادی می‌شد. علاوه بر این، فرآیند استخراج ویژگی‌ها و طبقه‌بندی ناحیه‌ها به صورت جداگانه انجام می‌شد که کارایی کلی مدل را کاهش می‌داد.

Fast R-CNN این مشکل را با اجرای عملیات کانولوشن برای کل تصویر به جای هر ناحیه پیشنهادی حل کرد و ویژگی‌های مشترک بین ناحیه‌های پیشنهادی را استخراج کرد، اما همچنان فرآیند تولید ناحیه‌های پیشنهادی (Region Proposals) خارج از شبکه و با استفاده از روش‌های سنتی مانند Selective Search انجام می‌شد که محدودیت سرعت را ایجاد می‌کرد.

این مشکلات باعث توسعه مدل‌هایی مانند Faster R-CNN شد که با معرفی شبکه‌ای به نام RPN (Region Proposal Network) فرآیند تولید ناحیه‌های پیشنهادی را به صورت یکپارچه در داخل شبکه انجام داد. بنابراین، محدودیت‌های سرعت و جداسازی مراحل پردازش درCNN ها دلیل اصلی توسعه این مدل‌های پیشرفته‌تر بوده است.

ه)

در شبکه‌های عصبی باقی‌مانده(ResNet) ، مفهوم اتصالات باقی‌مانده (Residual Connections) به گونه‌ای است که ورودی هر لایه یا بلوک مستقیماً به خروجی آن اضافه می‌شود. این ساختار باعث می‌شود که به جای یادگیری مستقیم نگاشت اصلی، شبکه تنها تغییرات یا "باقی‌مانده" بین ورودی و خروجی را مدل‌سازی کند. این رویکرد با فرمول $y=F(x)+x$ بیان می‌شود، که در آن x ورودی، $F(x)$ تابع یادگیری شده توسط لایه، و yyyخروجی است. اتصالات باقی‌مانده با فراهم کردن مسیر مستقیم برای انتقال گرادیان، مشکلاتی مانند ناپدید شدن گرادیان (Vanishing Gradient) را کاهش می‌دهند و یادگیری در شبکه‌های عمیق را تسهیل می‌کنند. علاوه بر این، این اتصالات یادگیری را بهینه‌تر می‌کنند زیرا شبکه تنها نیاز به مدل‌سازی تغییرات جزئی دارد. این معماری به شبکه امکان می‌دهد که عمیق‌تر شود (حتی تا ۱۵۲ لایه) بدون کاهش دقت یا مشکلات بهینه‌سازی، و دقت مدل را به طور چشمگیری افزایش می‌دهد. اتصالات باقی‌مانده به عنوان یک نوآوری کلیدی، امکان‌پذیری آموزش شبکه‌های بسیار عمیق را فراهم کرده و عملکرد مدل‌های یادگیری عمیق را بهبود بخشیده‌اند.

و)

برای مقابله با مسئله داده‌های کم در شبکه‌های عمیق، می‌توان از روش‌های مختلف regularization استفاده کرد. در ورودی داده‌ها، روش‌هایی مانند **افزایش داده‌ها (Data Augmentation)** با ایجاد تغییرات جزئی روی داده‌ها (مانند چرخش، تغییر مقیاس، برش، وارونگی یا تغییر روشنایی) و **نرمال‌سازی (Normalization)**برای مقیاس‌بندی ویژگی‌ها، تنوع داده‌ها را افزایش داده و عملکرد شبکه را بهبود می‌دهند. در ساختار شبکه، استفاده از **Dropout** برای غیرفعال کردن تصادفی نورون‌ها در هر مرحله، **نرمال‌سازی دسته‌ای (Batch Normalization)** برای تثبیت یادگیری، و **انتظام‌بخشی وزن‌ها (Weight Regularization)**با اعمال جریمه L1 یا L2 روی وزن‌ها، از پیچیدگی بیش از حد مدل جلوگیری می‌کند. در خروجی شبکه، روش **توقف زودهنگام (Early Stopping)** با متوقف کردن آموزش هنگام ثابت شدن عملکرد روی داده‌های اعتبارسنجی و استفاده از **روش‌های ترکیبی (Ensemble Methods)**برای ترکیب چندین مدل و افزایش دقت، از بیش‌برازش جلوگیری می‌کنند. این روش‌ها با توجه به نوع داده و مسئله می‌توانند تعمیم‌پذیری مدل را به طور مؤثری افزایش دهند.

سوال پنجم

Freezeکردن لایه‌های مدل به این معنا است که وزن‌ها و بایاس‌های لایه‌های اصلی مدل، که از پیش روی دیتاست ImageNet آموزش دیده‌اند، ثابت نگه داشته می‌شوند و در طول فرآیند آموزش تغییر نمی‌کنند. این روش باعث می‌شود که ویژگی‌های استخراج‌شده توسط این لایه‌ها برای طبقه‌بندی تصاویر جدید استفاده شوند، بدون نیاز به بازآموزی کامل مدل. این کار زمان آموزش را به میزان قابل‌توجهی کاهش می‌دهد و از نیاز به داده‌های زیاد برای یادگیری جلوگیری می‌کند. علاوه بر این، خطر بیش‌برازش (overfitting)نیز کاهش می‌یابد، زیرا لایه‌های اصلی از قبل بهینه شده‌اند. با این حال، در صورتی که داده‌های فعلی تفاوت قابل‌توجهی با داده‌های دیتاست ImageNet داشته باشند، ممکن است مدل نتواند عملکرد مطلوبی داشته باشد. بنابراین، برای تطبیق بهتر، تنها لایه‌های Fully Connected انتهایی آموزش داده می‌شوند تا مدل بتواند برای داده‌های خاص مسئله تنظیم شود.

تحلیل نتایج:

استفاده از مدل ResNet به‌عنوان یک مدل از پیش آموزش داده شده باعث شده است که ویژگی‌های عمومی تصاویر به خوبی استخراج شوند. این ویژگی‌ها که قبلاً روی دیتاست ImageNet آموزش دیده‌اند، به مدل اجازه داده‌اند که عملکرد مناسبی در طبقه‌بندی تصاویر COVID و Non-COVID داشته باشد. منجمد کردن لایه‌های اصلی مدل ResNet باعث کاهش زمان آموزش و جلوگیری از بیش‌برازش شده است، زیرا وزن‌های این لایه‌ها ثابت نگه داشته شده‌اند. با این حال، این رویکرد ممکن است مانع از یادگیری ویژگی‌های خاص‌تر داده‌های جدید شود. برای مثال، Recallبالای کلاس COVID (0.95) نشان می‌دهد که مدل برای این کلاس عملکرد خوبی داشته است، اما Recall پایین برای Non-COVID (0.68)نشان‌دهنده ضعف در تشخیص این کلاس است. این ضعف ممکن است به دلیل عدم انعطاف‌پذیری لایه‌های منجمد شده در استخراج ویژگی‌های خاص داده‌های Non-COVID باشد.

برای بهبود عملکرد مدل، می‌توان تعداد دوره‌های آموزشی (Epochs) را افزایش داد، زیرا نمودار دقت و ضرر نشان می‌دهد که مدل هنوز به طور کامل همگرا نشده است. افزایش تعداد دوره‌ها می‌تواند باعث بهبود یادگیری مدل شود. تنظیم نرخ یادگیری نیز مهم است؛ نرخ یادگیری فعلی (۰.۰۰۰۱) ممکن است برای همگرایی مدل بیش از حد کوچک باشد. پیشنهاد می‌شود که از نرخ یادگیری بالاتر مانند ۰.۰۰۱ استفاده شود و به تدریج کاهش یابد. همچنین استفاده از داده‌های بیشتر، به‌ویژه برای کلاس Non-COVID، می‌تواند عملکرد مدل را بهبود دهد. تکنیک‌های افزایش داده (Data Augmentation) مانند چرخش، برش، یا تغییر رنگ تصاویر نیز می‌توانند تنوع داده‌ها را افزایش داده و تعمیم‌پذیری مدل را بهتر کنند. در نهایت، اضافه کردن لایه‌های Fully Connected با تعداد نورون‌های بیشتر و Dropout برای کاهش بیش‌برازش نیز می‌تواند موثر باشد.

عدم توازن کلاس‌ها یکی از مشکلات اساسی در این مسئله است که باعث شده شبکه به سمت کلاس COVIDکه نمونه‌های بیشتری دارد سوگیری کند. این امر در Recall پایین برای کلاس-Non COVIDو عملکرد بهتر شبکه برای COVID قابل مشاهده است. برای حل این مشکل در سطح مجموعه داده، می‌توان از تکنیک‌های Oversampling برای افزایش نمونه‌های کلاس Non-COVID یا Undersamplingبرای کاهش نمونه‌های کلاس COVID استفاده کرد. در سطح تابع خطا، می‌توان از وزن‌دهی کلاس‌ها استفاده کرد؛ برای مثال، به کلاس Non-COVID وزن بیشتری اختصاص داده شود تا تأثیر عدم توازن کاهش یابد. همچنین، استفاده از روش‌هایی مانند Focal Loss که نمونه‌های دشوارتر را بیشتر وزن‌دهی می‌کند، می‌تواند به تعادل شبکه کمک کند. علاوه بر این، استفاده از متریک‌هایی مانند F1-Scoreبه جای دقت (Accuracy) برای ارزیابی عملکرد مدل توصیه می‌شود، زیرا این متریک‌ها تأثیر عدم توازن کلاس‌ها را کاهش می‌دهند.

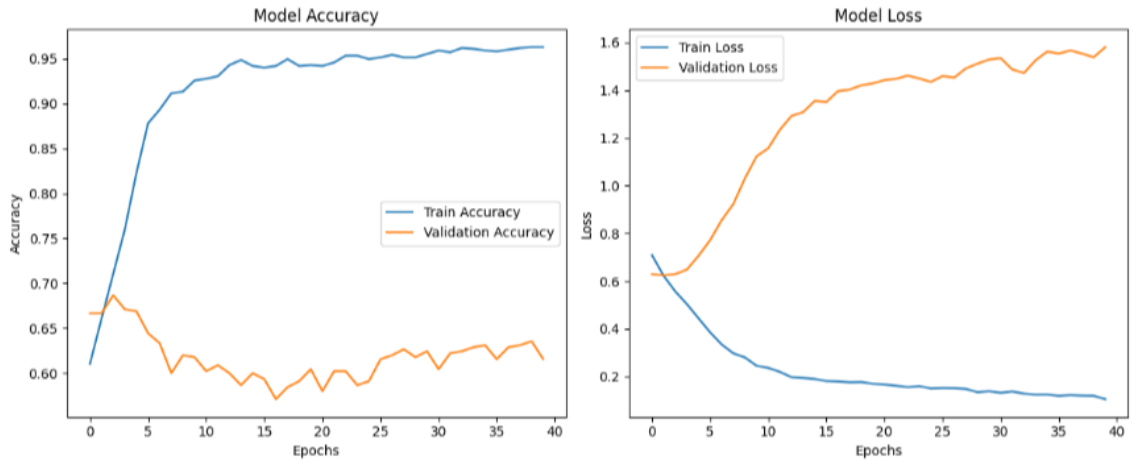
گزارش نهایی:

این مدل بر اساس معماری ResNet101 طراحی شده است که یک شبکه عصبی عمیق و از پیش آموزش‌دیده روی دیتاست ImageNet است ResNet101 .به دلیل استفاده از اتصالات باقیمانده مشکلاتی نظیر ناپدید شدن گرادیان در شبکه‌های عمیق را حل می‌کند. لایه‌های اصلی این مدل فریز شده‌اند تا ویژگی‌های استخراج‌شده از قبل حفظ شوند و ویژگی‌های جدیدی با داده‌های فعلی یاد گرفته نشود. لایه‌های کاملاً متصل جدیدی برای تطبیق مدل با مسئله طبقه‌بندی دودویی (COVID و-Non COVID)اضافه شده‌اند. لایه GlobalAveragePooling2D برای کاهش ابعاد نقشه ویژگی‌های خروجی ResNet101به کار گرفته شد. یک لایه Dense با ۲۵۶ نورون و تابع فعال‌سازی ReLU برای یادگیری روابط پیچیده اضافه شد Dropout .با نرخ ۰.۵ برای جلوگیری از بیش‌برازش به مدل اضافه شد و سپس یک Dense با ۱۲۸ نورون همراه با تابع ReLU و Dropout با نرخ ۰.۳ به مدل افزوده شدند. لایه خروجی مدل یک Dense با یک نورون و تابع فعال‌سازی Sigmoid بود که برای طبقه‌بندی دودویی استفاده می‌شود.

فریز کردن لایه‌های ResNet101 باعث می‌شود وزن‌ها و بایاس‌های این لایه‌ها در طول آموزش تغییر نکنند. این کار به مدل اجازه می‌دهد از ویژگی‌های استخراج‌شده که قبلاً آموزش داده شده‌اند، بدون نیاز به بازآموزی، استفاده کند. فریز کردن به ویژه در دیتاست‌های کوچک یا زمانی که ویژگی‌های استخراج‌شده مدل اصلی با مسئله فعلی تطابق دارند، مؤثر است. این روش زمان آموزش را کاهش می‌دهد، نیاز به داده را کم می‌کند و خطر بیش‌برازش را کاهش می‌دهد.

در تنظیمات هایپرپارامترها، از Binary Crossentropy به عنوان تابع هزینه استفاده شده است که برای مسائل طبقه‌بندی دودویی مناسب است Adam .به عنوان بهینه‌ساز انتخاب شده است که به دلیل نرخ یادگیری تطبیقی و کارایی در پردازش گرادیان‌های پراکنده مؤثر است. نرخ یادگیری مقدار ۰.۰۰۰۱ انتخاب شده است تا همگرایی پایدار و جلوگیری از پرش به مقادیر نادرست تضمین شود.

نمودارهای دقت و ضرر برای داده‌های آموزش و اعتبارسنجی طی ۴۰ دوره رسم شده‌اند. دقت آموزش به سرعت به حدود ۹۵ درصد رسید اما دقت اعتبارسنجی بهبود آهسته‌تری داشت و در حدود ۷۰ درصد ثابت شد. ضرر آموزش به طور پیوسته کاهش یافت که نشان‌دهنده یادگیری مؤثر است، اما ضرر اعتبارسنجی با گذشت زمان افزایش یافت که احتمالاً نشان‌دهنده بیش‌برازش است.

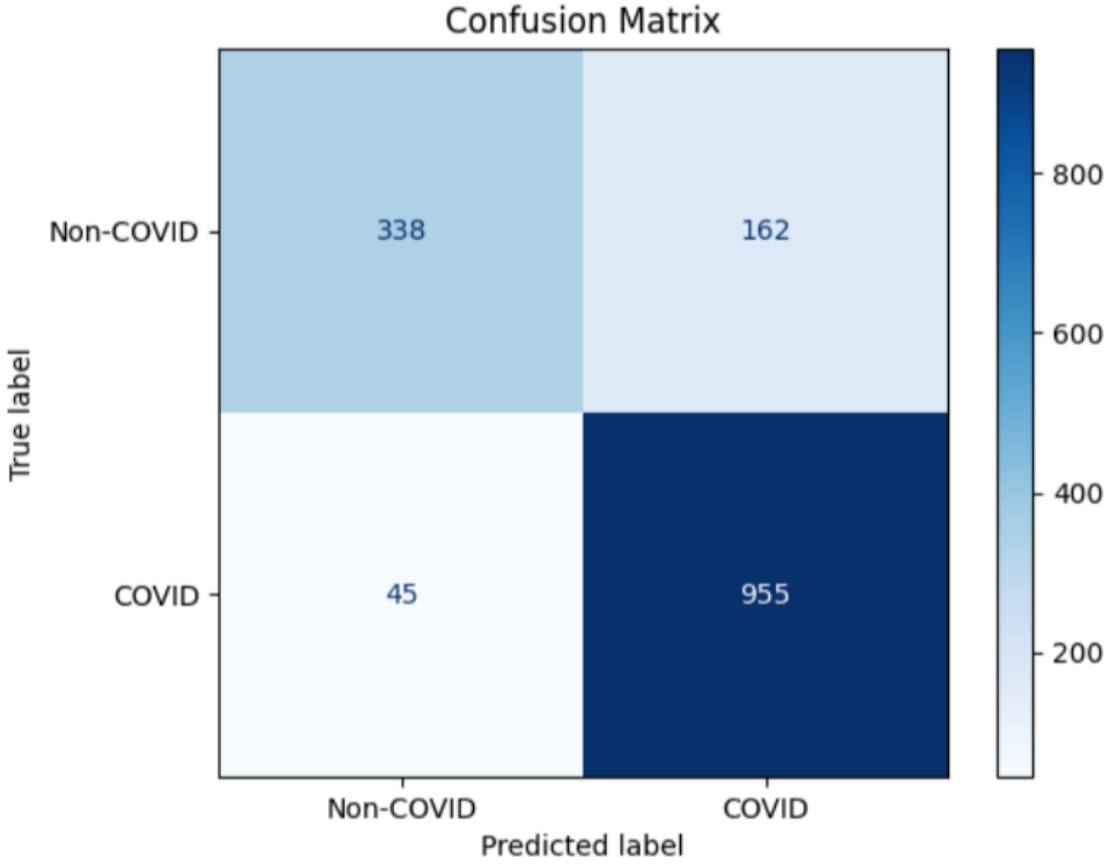


جدول ارزیابی عملکرد مدل به شرح زیر است:

کلاس	دقت(Precision)	بازیابی(Recall)	امتیاز F1	تعداد نمونه(Support)
Non-COVID	0.88	0.68	0.77	500
COVID	0.85	0.95	0.90	1000
دقت کلی			**0.86**	1500
میانگین کل	0.87	0.82	0.83	1500
میانگین وزنی	0.86	0.86	0.86	1500

مدل دقت کلی ۸۶ درصد را به دست آورده است. دقت پیش‌بینی برای COVID برابر ۸۵ درصد و برای Non-COVIDبرابر ۸۸ درصد بود. بازیابی برای COVID برابر ۹۵ درصد و برای Non-COVID برابر ۶۸ درصد بود. امتیاز F1 برای COVID برابر ۹۰ درصد و برای Non-COVID برابر ۷۷ درصد محاسبه شد.

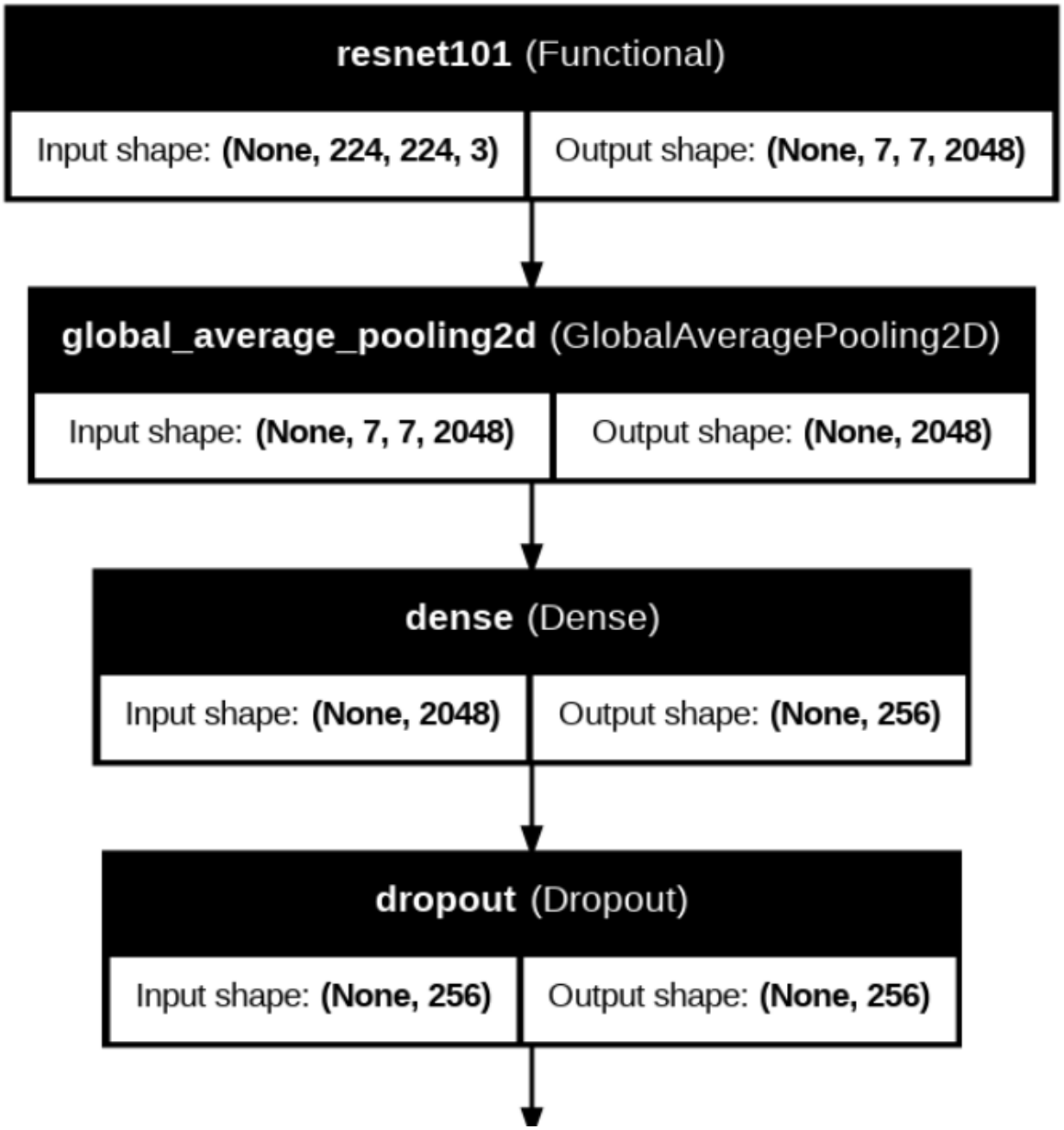
میانگین کل برای Precision برابر ۸۷ درصد، Recallبرابر ۸۲ درصد و F1 برابر ۸۳ درصد است. میانگین وزنی برای Precision برابر ۸۶ درصد، Recallبرابر ۸۶ درصد و F1 برابر ۸۶ درصد بود.

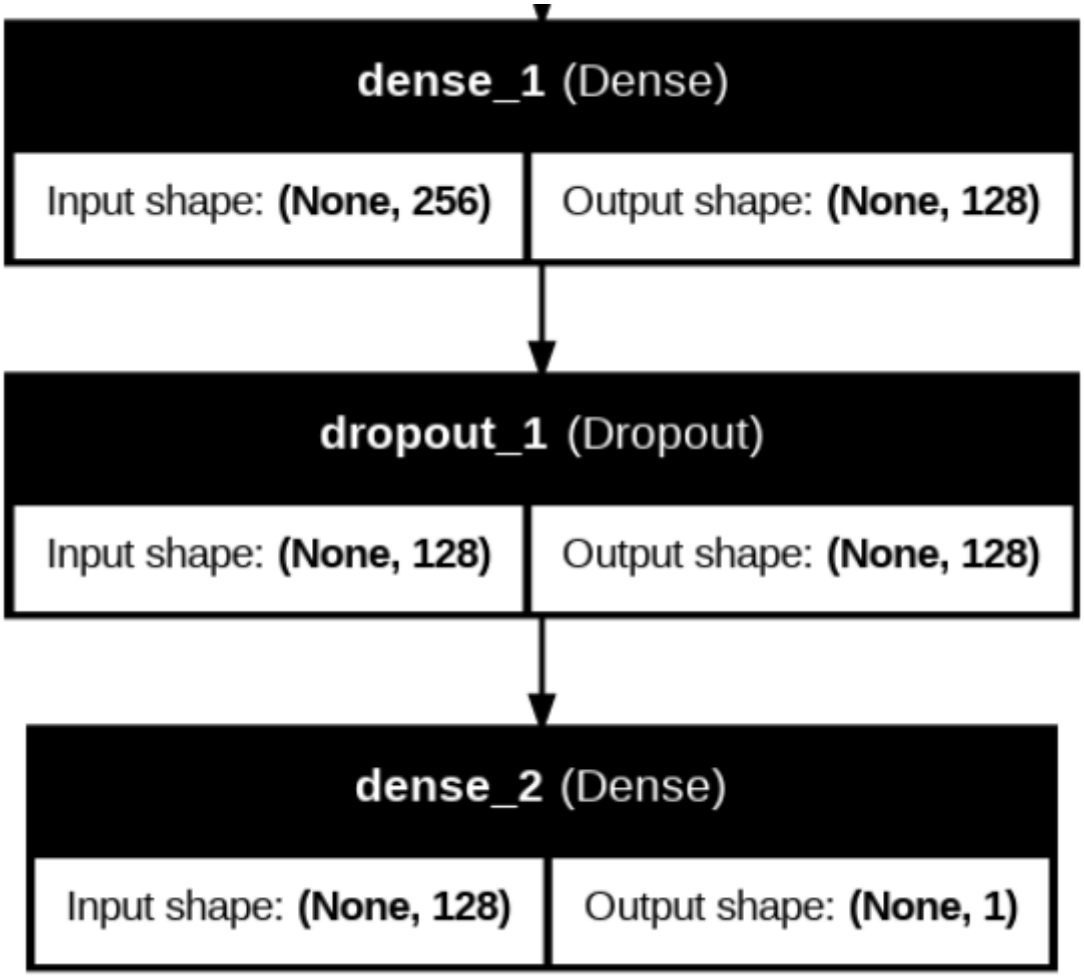


ماتریس آشفتگی نشان داد که مدل ۹۵۵ نمونه COVID را درست پیش‌بینی کرده و ۴۵ نمونه را به اشتباه Non-COVID تشخیص داده است. همچنین مدل ۳۳۸ نمونه Non-COVID را درست پیش‌بینی کرده و ۱۶۲ نمونه را به اشتباه COVID تشخیص داده است.

عملکرد مدل نشان می‌دهد که در تشخیص موارد COVID عملکرد بالایی دارد، اما بازیابی پایین در کلاس Non-COVID نشان‌دهنده نرخ بالای False Negative در این کلاس است. این عدم توازن می‌تواند به دلیل نامتعادلی داده‌ها یا سوگیری مدل نسبت به کلاس COVID به دلیل حجم بالاتر نمونه‌ها باشد. افزایش داده‌ها، استفاده از تکنیک‌های منظم‌سازی بیشتر و تنظیم لایه‌های پایه می‌تواند عملکرد مدل را بهبود بخشد.

معماری مدل:


resnet101 (Functional)
Input shape: (None, 224, 224, 3)
Output shape: (None, 7, 7, 2048)
global_average_pooling2d (GlobalAveragePooling2D)
Input shape: (None, 7, 7, 2048)
Output shape: (None, 2048)
dense (Dense)
Input shape: (None, 2048)
Output shape: (None, 256)
dropout (Dropout)
Input shape: (None, 256)
Output shape: (None, 256)



سوال ششم(امتیازی)