فهرست

شکل‌ها

جدول‌ها

پرسش۱. طبقه‌بندی تصاویر با VIT

تمام کدهای این سوال و نمودارها و نتایج حاصل در فایل VIT.ipynb واقع در پوشه appendix در مسیر اصلی فولدر موجود می‌باشد.

۱-۱- مقدمه

۱-۱-۱- اصلی‌ترین تفاوت و مزیت مدل‌های ویژن ترنسفورمر با مدل‌های سنتی

تفاوت اصلی: مدل‌های CNN تصاویر را به صورت محلی پردازش می‌کنند یعنی تمرکز روی نواحی کوچک تصویر می‌کنند و ویژگی‌ها را به صورت سلسله مراتبی و از پایین به بالا استخراج می‌کنند. اما VIT تصویر ار به قطعات یا پچ‌های کوچک تقسیم می‌کند و هر پچ را مشابه یک توکن در NLP در نظر می‌گیرد، سپس با استفاده از مکانیزم توجه(self-attention) وابستگی بین تمام پچ‌ها را در نظر میگیرد.

مزیت VIT نسبت به CNN:

اصلی ترین مزیت مدل نسبت به روشهای سنتی برخورداری از توانایی یادگیری وابستگی‌های سراسری و پیچیده بر پایه self-attention است و طبق نتایج مقاله، مدل ViT در مقایسه با مدل‌های CNN (مثلResNet ، MobileNet وEfficientNet) دقت بالاتری(تا ۹۹.۸٪) در شناسایی بیماری‌ها از روی تصاویر گیاهان ارائه می‌دهد.

Global Context: برخلاف CNN که دید محدودی دارد، VIT می‌تواند رابطه بین بخش‌های مختلف تصویر را بهتر تشخیص دهد، که برای شناسایی بیماری‌هایی که الگوهای پراکنده دارند بسیار مفید است.

عملکرد بهتر روی داده‌های پیچیده: VIT توانایی بیشتری در یادگیری ویژگی‌های پیچیده و غیرمحلی دارد، به همین دلیل در تشخیص دقیق‌تر انواع بیماری‌های گیاهی عملکرد بهتری دارد.

عدم نیاز به طراحی دستی ویژگی‌ها: در CNN معمولا برای بهینه‌سازی مدل باید معماری‌های خاص طراحی شود. اما VIT با ساختار ساده‌تری و بدون نیاز به طراحی خاص لایه‌ها نتایج بسیار خوبی ارائه می‌دهد.

۱-۱-۲- مدل بهتر در شرایطی که دادگان محدود است

در شرایطی که حجم داده‌های آموزشی محدود است، مدل‌های سنتی مانند CNN (مثلاً ResNet یا MobileNet) عملکرد بهتری نسبت به ViT دارند.

CNN با استفاده از فیلترهای محلی، ویژگی‌ها را به صورت گام‌به‌گام از تصویر استخراج می‌کند، که باعث می‌شود حتی با داده‌های نسبتاً کم بتواند ویژگی‌های مفید را بیاموزد. اما ViT ساختار بسیار

انعطاف‌پذیری دارد و فاقد inductive bias (سوگیری‌های ذاتی در پردازش تصویر مانند همسایگی محلی و اشتراک وزن‌ها) است. این بدان معناست که برای یادگیری مفید، نیاز به داده‌های بسیار زیاد دارد تا بتواند وابستگی‌ها و الگوها را کشف کند.

Self-Attention با همه‌ی پچ‌های تصویر کار می‌کند و در نبود داده کافی، به‌سختی می‌تواند وابستگی معنادار بین نواحی مختلف تصویر را بیاموزد. این باعث می‌شود ViT در داده‌های کوچک دچار overfittin یا generalization ضعیف شود.

در مقاله، برای حل مشکل کمبود داده در Vit، نویسندگان از روش زیر استفاده کردند:

پیش‌آموزش (Pretraining) مدل روی داده‌های بزرگ‌تر، و سپس انتقال یادگیری (Transfer Learning) به مجموعه‌ی کوچک‌ترPlantVillage. این تکنیک کمک کرد تا ViT بتواند با وجود داده‌های محدود، همچنان عملکرد بسیار خوبی داشته باشد.

۱-۲- آماده‌سازی داده‌ها

۱-۲-۱- بررسی توازن دادگان

جدول ۱. توازن دادگان

لیبل(label)	نام بیماری(disease_name)	تعداد دادگان(count)
۰	Tomato__Target_Spot	۱۰۰۰
۱	Tomato_Spider_mites	۱۰۰۰
۲	Tomato_Bacterial_spot	۱۰۰۰
۳	Tomato_Late_blight	۱۰۰۰
۴	Tomato_Septoria_leaf_spot	۱۰۰۰
۵	Tomato_Leaf_Mold	۹۵۲
۶	Tomato_Early_blight	۱۰۰۰
۷	Tomato_YellowLeaf_Curl_Virus	۱۰۰۰
۸	Tomato_healthy	۱۰۰۰
۹	Tomato_mosaic_virus	۳۷۳

خیر داده‌ها متوازن نیستند. کلاس شماره ۹ (Tomato_mosaic_virus) با تنها ۳۷۳ تصویر به‌وضوح نسبت به بقیه کلاس‌ها (که حدود ۱۰۰۰ تصویر دارند) داده‌ی بسیار کمتری دارد. کلاس ۵ نیز (با ۹۵۲ تصویر) تا حدی کمتر است ولی قابل چشم‌پوشی است.

برای افزایش داده کلاس‌های دارای تصاویر کمتر (به‌ویژه کلاس ۹)، می‌توان از تکنیک‌های ساده اما مؤثر استفاده کرد، که ساختار ظاهری بیماری را تخریب نکند مثل:

اعمال چرخش‌های محدود (تا ۱۰ درجه)، تغییرات روشنایی، کنتراست و اشباع خفیف، جابه‌جایی‌های کوچک (Affine) و محوکردن ملایم (Gaussian Blur) همگی از جمله روش‌هایی هستند که باعث ایجاد تنوع ظاهری در تصاویر می‌شوند، در حالی که ساختار اصلی برگ و نشانه‌های بیماری حفظ می‌شود. فلیپ افقی (با احتمال ٪۵۰) نیز با توجه به ساختار تقریباً متقارن برگ‌های گوجه‌فرنگی بی‌ضرر است. همچنین احتمال پایین برای فلیپ عمودی (٪۲۰) و استفاده از زوم یا برش تصادفی کنترل‌شده باعث شبیه‌سازی شرایط مختلف تصویربرداری می‌شود. این تکنیک‌ها با هدف افزایش نمونه‌های کلاس‌های نادر (مانند ویروس موزاییک گوجه‌فرنگی) به کار می‌روند و از آنجا که ویژگی‌های کلیدی تصاویر، مانند شکل لکه‌ها یا الگوهای بیماری، حفظ می‌شوند، این نوع تقویت داده در متعادل‌سازی کلاس‌ها بسیار مؤثر و از نظر علمی موجه است.

۱-۲-۲- پیش‌پردازش‌های دیگر

برای بهبود عملکرد مدل و کاهش overfitting ، علاوه بر Resize و ToTensor ، از تکنیک‌های Data Augmentation نظیر RandomHorizontalFlip، RandomRotation و ColorJitter استفاده شد. همچنین، تصاویر ورودی طبق استاندارد ImageNet نرمالایز شدند (mean=[0.485,0.456,0.406] و std=[0.229,0.224,0.225]). این پیش‌پردازش‌ها هم در مدل CNN (Inception-V3) و هم در مدل ViTاعمال شدند.

۱-۳- آموزش مدل CNN

۱-۳-۱- شرح نحوه کارکرد کلی مدل Inception-v3

جدول ۲پارامترهای مورد استفاده در مدل CNN

تعداد خروجی	نرخ یادگیری	Batch size	تعداد ایپاک	بهینه‌ساز	تابع هزینه
۱۰=۹کلاس + سالم	۰.۰۰۱	۳۲	۳۰	Adam	Cross-Entropy Loss

Inception-v3یک شبکه عصبی کانولوشنی (CNN) عمیق است که هدف اصلی آن استخراج ویژگی‌های پیچیده و چندسطحی از تصاویر به منظور دسته‌بندی (classification) است. مدل با دریافت یک تصویر رنگی RGB در اندازه استاندارد (معمولاً ۲۹۹x۲۹۹ پیکسل) شروع می‌کند. تصویر معمولاً پیش‌پردازش می‌شود، شامل تغییر اندازه، نرمال‌سازی (Normalization) با میانگین و انحراف معیار ثابت (مثلاً ImageNet) تا مقیاس پیکسل‌ها برای مدل مناسب شود. سپس وارد مرحله استخراج ویژگی می‌شود و تصویر وارد چندین لایه کانولوشنی (Convolutional layers) می‌شود که هر کدام فیلترهایی دارند تا الگوهای ابتدایی مثل لبه‌ها، بافت‌ها و شکل‌ها را تشخیص دهند. درInception-v3 ، این مرحله به صورت ماژولار انجام می‌شود؛ هر ماژول Inception ترکیبی از چند شاخه با فیلترهای متفاوت (مثل ۱x۱، ۳x۳، ۵x۵) است که به صورت همزمان بر روی تصویر اعمال می‌شوند. این کار باعث می‌شود مدل بتواند ویژگی‌ها را در چند مقیاس مختلف شناسایی کند. بعد مدل به تدریج ابعاد فضایی داده‌ها را کاهش می‌دهد (مثلاً با لایه‌هایPooling) ولی تعداد کانال‌ها و ویژگی‌ها را افزایش می‌دهد تا اطلاعات بیشتر و پیچیده‌تری استخراج شود در این مرحله کاهش ابعاد و تجمیع اطلاعات صورت میگیرد. در انتهای شبکه، تمام ویژگی‌های استخراج شده به یک لایه کاملاً متصل (FC) داده می‌شوند این لایه FC نقش «کلاسه‌بند» را دارد و خروجی آن یک بردار با طول برابر تعداد کلاس‌های مسئله است (مثلاً ۱۰ کلاس). خروجی لایه FC معمولاً از طریق تابع SoftMax عبور می‌کند تا احتمال تعلق تصویر به هر کلاس محاسبه شود. مدل پیش‌بینی می‌کند که تصویر متعلق به کدام کلاس با بالاترین احتمال است. و جهت آموزش مدل با داده‌های برچسب‌خورده و به کمک تابع هزینه (مثلاًCross-Entropy Loss) آموزش داده می‌شود و وزن‌های فیلترها و لایه‌ها به مرور و بر اساس گرادیان‌ها تنظیم می‌شوند تا دقت تشخیص افزایش یابد.

به طور خلاصه:

Inception-v3 تصویر را در چند مقیاس تحلیل می‌کند (ماژول‌هایInception).

ویژگی‌های پیچیده استخراج شده به یک کلاسه‌بند داده می‌شوند.

خروجی احتمال تعلق تصویر به هر کلاس را می‌دهد.

مدل به کمک داده‌های برچسب‌خورده یاد می‌گیرد چطور بهتر ویژگی‌ها را استخراج و طبقه‌بندی کند.

۱-۳-۲- شرح تابع خطا استفاده شده درCNN

در اکثر مسائل دسته‌بندی چندکلاسه، از تابع خطا یا loss function به نام Cross Entropy Loss استفاده می‌کنند. در اینجا نیز از همین تابع استفاده شده است، این تابع خطا برای دسته‌بندی چندکلاسه بسیار استاندارد است.

فرض کنید مدل برای هر نمونه، یک بردار از نمرات (logits) کلاس‌ها تولید می‌کند این نمرات با تابع SoftMax به احتمال تعلق نمونه به هر کلاس تبدیل می‌شوند. Cross Entropy Loss اختلاف بین احتمال‌های پیش‌بینی شده و برچسب واقعی (که به صورت one-hot کد شده) را محاسبه می‌کند و هدف بهینه‌سازی، کم کردن این خطا است که یعنی مدل احتمال کلاس درست را افزایش دهد.

$$Loss = -\sum_{c=1}^{C} y_c \log(\hat{y}_c)$$

که در آن C تعداد کلاس‌هاست

y_c مقدار واقعی کلاس (۰ یا ۱)

$\hat{y}_c$ احتمال پیش‌بینی شده برای کلاس c است.

توابع دیگر:

بسته به کاربرد و شرایط ممکن است توابع خطای دیگری هم استفاده شود:

الف)Focal Loss

وقتی داده‌ها نامتوازن (imbalanced) باشند، این تابع کمک می‌کند که نمونه‌های سخت‌تر یا کمتر دیده شده وزن بیشتری داشته باشند.

مخصوصاً در مسائل دسته‌بندی که برخی کلاس‌ها کمتر دیده شده‌اند مفید است.

ب)Label Smoothing

این یک تکنیک اصلاحی روی Cross Entropy است که برچسب‌های سخت (۰ و ۱) را کمی نرم می‌کند (مثلاً ۰.۹ و ۰.۱).

این باعث جلوگیری از overfitting و افزایش تعمیم‌پذیری مدل می‌شود.

ج)Hinge Loss

بیشتر در مسائل دسته‌بندی دوکلاسه (SVM) کاربرد دارد، ولی نسخه‌های چندکلاسه هم دارد.

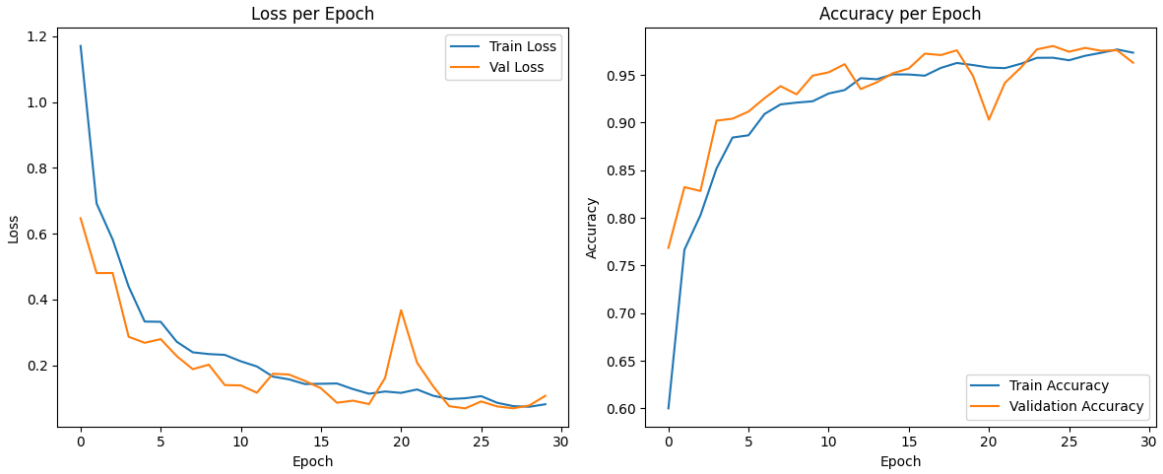
کمتر برایCNN های عمیق استفاده می‌شود.

د)Kullback-Leibler Divergence Loss (KL Divergence)

اگر بخواهید مدل را با خروجی احتمالات هدف که نرم‌تر هستند (مثل مدل‌های ensemble یا distillation) آموزش دهید، KL Divergence مناسب است.

مثلا اگر دیتاست نامتوازن است، می‌توانیم Focal Loss را امتحان کنیم یا برای بهبود تعمیم‌پذیری می‌توانید از Label Smoothing همراه Cross Entropy استفاده کنید.

۱-۳-۳- نمودار دقت و خطا و ماتریس آشفتگی مدل CNN

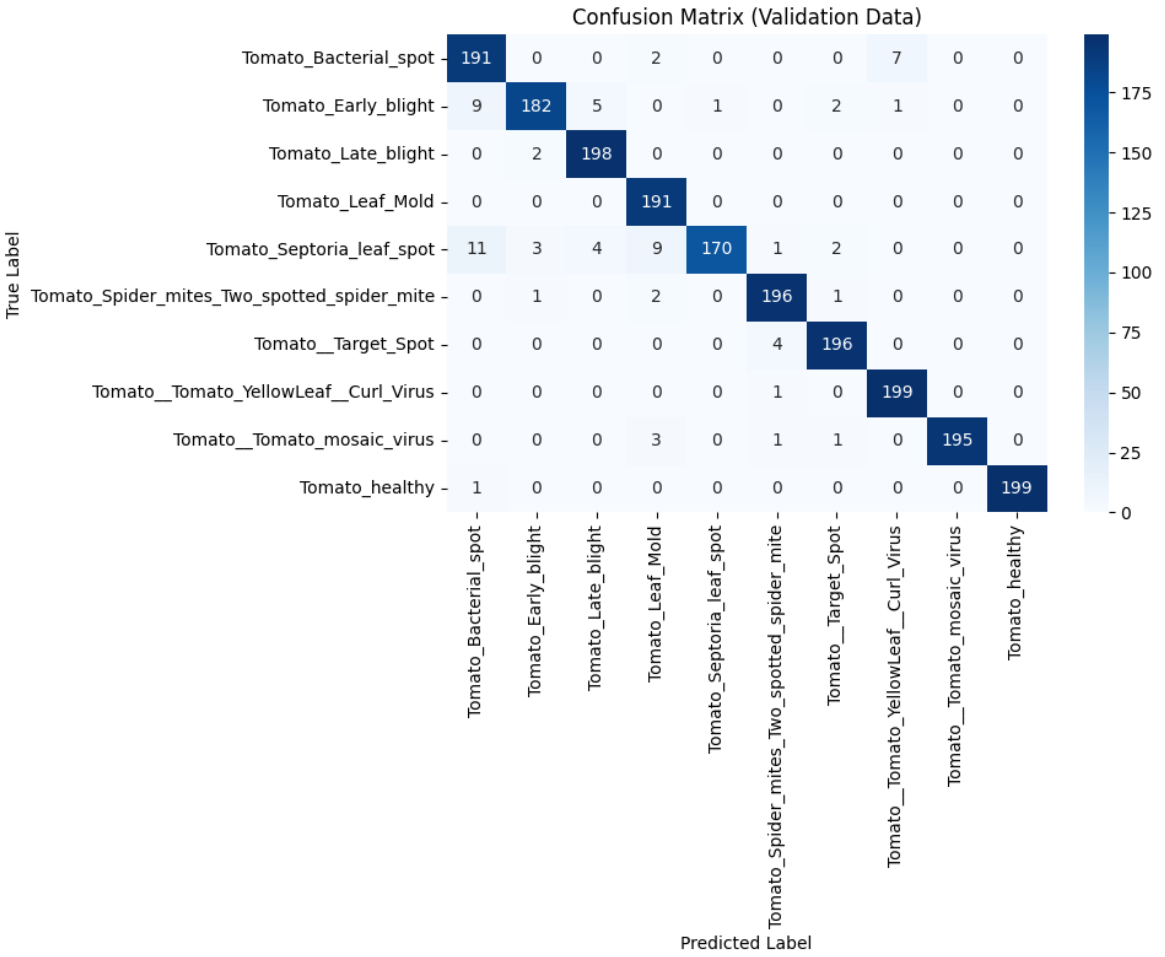


شکل ۱. نمودار دقت و خطا داده آموزش و اعتبارسنجی در CNN

بررسی نمودارهای آموزش و اعتبارسنجی (شکل ۱. نمودار دقت و خطا داده آموزش و اعتبارسنجی در CNN) نشان می‌دهد که مدل از همان ایپاک‌های ابتدایی (ایپاک ۱ تا ۵) روندی سریع در کاهش خطا (Loss) و افزایش دقت (Accuracy) داشته است؛ به‌طوری که دقت آموزش از حدود ٪۶۰ در ایپاک اول به بیش از ٪۸۰ در ایپاک سوم رسید و دقت اعتبارسنجی نیز از ٪۷۶ به حدود ٪۸۳ افزایش یافت. این روند نشان‌دهنده توانایی مدل در یادگیری ویژگی‌های اولیه تصاویر بیماری‌هاست.

در ادامه، در ایپاک‌های میانی (ایپاک ۱۰ تا ۲۰)، مدل به مرحله‌ای از پایداری نسبی رسید و در این بازه، دقت آموزش به ٪۹۲ و بالاتر و دقت اعتبارسنجی به حدود ٪۹۵ رسید. نمودارهای دقت در این مرحله نشان می‌دهد که شیب افزایش دقت کاهش یافته و مدل در حال همگرایی به وضعیت مطلوب بوده است. جالب است که در ایپاک ۲۱، برخلاف روند کاهشیLoss ، ناگهان Loss اعتبارسنجی افزایش پیدا کرده (به ۰.۳۶۷۴) و دقت اعتبارسنجی افت محسوسی داشته است (به ٪۹۰.۳۱). این نوسان می‌تواند به دلیل وجود داده‌های Validation چالش‌برانگیزتر یا بروز پدیده‌ی overfitting در آن ایپاک

باشد. با این حال، مدل مجدداً در ایپاک‌های بعدی توانست Loss را کاهش دهد و دقت را بازیابی کند؛ به‌طوری که در ایپاک ۲۵ به دقت ٪۹۸.۰۴ در اعتبارسنجی رسید و در ایپاک ۳۰ نیز عملکرد مطلوبی (٪۹۶.۲۸) داشت.



شکل ۲. ماتریس آشفتگی مدل CNN

ماتریس آشفتگی (شکل ۲. ماتریس آشفتگی مدل CNN) اطلاعات ارزشمندی درباره‌ی نقاط قوت و ضعف مدل ارائه می‌دهد. برای مثال، در تشخیص کلاس‌های Tomato Mosaic Virus و Tomato Yellow Leaf Curl Virus، مدل تقریباً بدون خطا عمل کرده است و پیش‌بینی‌ها کاملاً منطبق با برچسب‌های واقعی بوده‌اند. اما در مقابل، کلاس Tomato Septoria Leaf Spot دچار بیشترین آشفتگی شده و به‌ویژه با کلاسTomato Leaf Mold اشتباه گرفته شده است. این موضوع احتمالاً به دلیل شباهت ظاهری برخی علائم بیماری‌ها در این دو کلاس است. همچنین، کلاس Tomato Early Blight نیز در چند مورد (۹ تصویر) با کلاس‌های مشابه اشتباه شده که می‌تواند به داده‌های چالش‌برانگیز یا ویژگی‌های بصری مشترک بین این بیماری‌ها برگردد.

در نهایت، بررسی کلی مدل بیانگر این است که Inception-V3 توانسته است با دقت بالایی (بیش از ٪۹۷) بیماری‌های برگ گوجه‌فرنگی را تشخیص دهد. روند کاهش Loss و افزایش Accuracy در نمودارها به‌خوبی این موضوع را نشان می‌دهد و ماتریس آشفتگی نیز کمک کرده است تا کلاس‌های نیازمند بهبود شناسایی شوند.

۱-۴- آموزش مدل Vit

۱-۴-۱- ملاحظات سخت‌افزاری

در این تمرین، مدل ViT بر اساس ساختار کلی ارائه‌شده در مقاله "Vision Transformer-Based Tomato Disease Classification" پیاده‌سازی شده است. ساختار مدل از نظر اندازه تصویر ورودی (۶۴×۶۴ پیکسل)، اندازه پچ‌ها (۸×۸)، تعداد توکن‌ها (۶۴)، ابعاد embedding (۶۴)، و تعداد بلوک‌های ترنسفورمر (۸) دقیقاً مطابق مقاله تنظیم شده است. با این حال، یک تفاوت مهم در نحوه استخراج خروجی نهایی وجود دارد.

در مدل مقاله (مطابق جدول ۴ مقاله)، پس از لایه‌های attention ، تمام توکن‌ها (توالی ۱۰۰×۶۴ شامل پچ‌ها و cls token) به‌صورت کامل flatten شده و وارد چندین لایه بسیار بزرگ Dense (با ابعاد ۲۰۴۸ و ۱۰۲۴) شده‌اند. این ساختار منجر به افزایش چشم‌گیر تعداد پارامترها (بیش از ۱۷ میلیون فقط در لایه‌های Dense نهایی) و همچنین بار محاسباتی سنگین‌تر می‌شود.

در مقابل، در پیاده‌سازی حاضر از ساختار مرسوم ViT استفاده شده و تنها خروجی cls_token به‌عنوان نماینده کل تصویر برای طبقه‌بندی نهایی به لایه خروجی متصل شده است. این انتخاب باعث کاهش قابل توجه تعداد پارامترها، مصرف حافظه و زمان آموزش شده، در حالی که دقت مدل همچنان در سطح قابل قبولی (حدود ٪۸۷) باقی مانده است. این تغییر با هدف افزایش بهره‌وری محاسباتی و تطابق با محدودیت سخت‌افزاری موجود صورت گرفته و ساختار اصلی ترنسفورمر و attention همچنان حفظ شده است.

۱-۴-۲- توضیحات در مورد لایه Patch Embedding

لایه Patch Embedding یکی از بنیادی‌ترین اجزای شبکه‌های Vision Transformer (ViT) است. و برای تبدیل تصویر به توالی پچ‌های برداری‌شده جهت ورود به ترنسفورمر استفاده می‌شود.

در شبکه‌های ViT ، به‌جای استفاده مستقیم از کل تصویر، تصویر ابتدا به تکه‌هایی (patches) تقسیم می‌شود. سپس هر patch مانند یک "کلمه" در NLP به یک بردار عددی (embedding) تبدیل می‌شود.

فرض کنید تصویر ورودی اندازه‌اش ۳×۲۲۴×۲۲۴ باشد.

مثلاً اگر پچ برابر ۱۶ باشد، تصویر به ۲۲۴/۱۶ x ۲۲۴/۱۶ = ۱۹۶ پچ تقسیم می‌شود.

هر patch با ابعاد۳ ×۱۶×۱۶ به یک بردار با طول ثابت (مثلاً ۷۶۸) فشرده می‌شود با یک لایه linear یا convolution.

تصویر تبدیل می‌شود به دنباله‌ای از ۱۹۶ بردار با طول ۷۶۸، یعنی یک ورودی مشابه توالی درNLP .

این تبدیل، تصاویر را به توالی بردارها تبدیل می‌کند تا بتوان آن‌ها را به لایه‌های ترنسفورمر داد چون ترنسفورمر اساساً برای توالی‌ها طراحی شده.

تاثیر اندازه پچ در این تمرین:

جدول ۳. تاثیر اندازه patch

اندازه patch	مزایا
افزایش اندازه پچ (مثلاً از ۸×۸ به ۱۶×۱۶)	تعداد پچ‌ها کاهش می‌یابد ⟶ مدل جزئیات محلی کمتری می‌بیند ولی سریع‌تر است.
کاهش اندازه پچ (مثلاً از ۸×۸ به ۴×۴)	تعداد پچ‌ها زیاد می‌شود ⟶ مدل دقت بیشتری در تشخیص جزئیات دارد ولی محاسباتی‌تر و کندتر است.

در مدل‌هایVision Transformer ، برخلاف شبکه‌های کانولوشنی که به‌صورت پیوسته تصویر را پردازش می‌کنند، ابتدا تصویر به تکه‌های کوچکتری به نام «پچ» تقسیم می‌شود و هر پچ به‌نوعی مثل یک توکن در مدل زبانی عمل می‌کند. به‌همین دلیل، اندازه‌ی این پچ‌ها نقش مهمی در نحوه‌ی یادگیری مدل ایفا می‌کند.

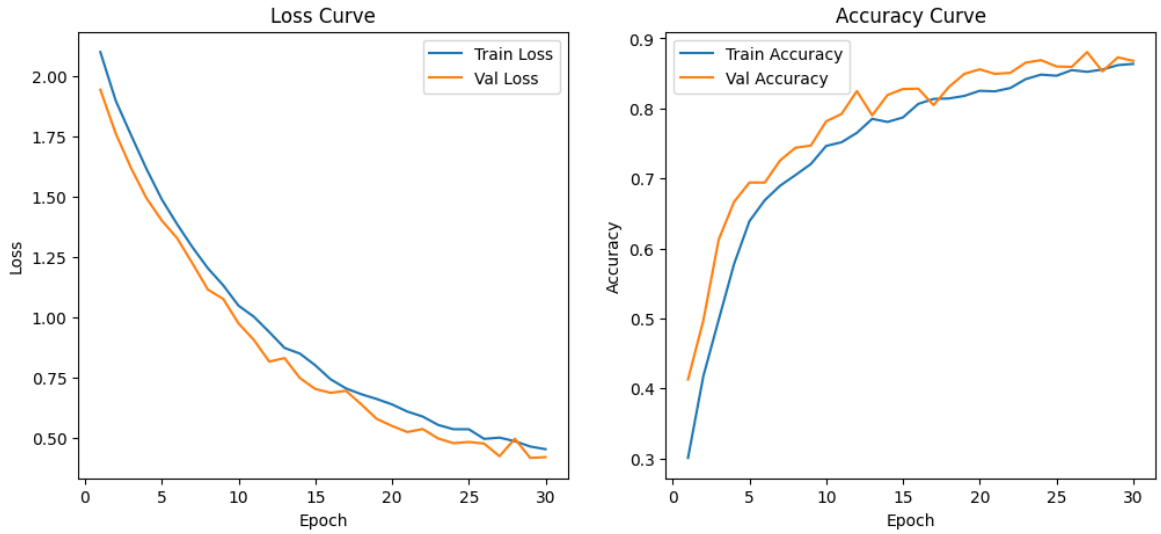
در این تمرین، تصاویر ورودی با ابعاد ۶۴×۶۴ پیکسل به مدل داده شده‌اند و اندازه‌ی هر پچ برابر با ۸×۸ در نظر گرفته شده است. این یعنی تصویر به ۶۴ بخش کوچک‌تر تقسیم می‌شود. این عدد نه آن‌قدر زیاد است که محاسبات مدل را سنگین کند، و نه آن‌قدر کم که مدل نتواند جزئیات تصویر را تشخیص دهد. به‌عبارتی، این انتخاب یک تعادل خوب بین دقت و کارایی ایجاد کرده است.

برای درک بهتر، اگر اندازه‌ی پچ را کوچکتر کنیم (مثلاً ۴×۴)، تعداد پچ‌ها به ۲۵۶ می‌رسد. این باعث می‌شود مدل بتواند با دقت بیشتری ویژگی‌های ظریف تصویر مثل لکه‌های کوچک یا تغییر رنگ‌های جزئی روی برگ را تشخیص دهد. اما از طرف دیگر، به‌دلیل افزایش قابل توجه تعداد توکن‌ها، هم سرعت آموزش پایین‌تر می‌آید و هم نیاز به حافظه و توان پردازشی بیشتری خواهد بود.

در مقابل، اگر اندازه‌ی پچ را بزرگ‌تر کنیم (مثلاً ۱۶×۱۶ که فقط ۱۶ پچ می‌دهد)، سرعت پردازش بیشتر می‌شود، اما مدل بخشی از جزئیات مهم بیماری را از دست خواهد داد؛ چون ممکن است چند ویژگی مهم داخل یک پچ ترکیب شوند و تفکیک‌پذیری مدل کمتر شود.

بنابراین، انتخاب اندازه‌ی ۸×۸ در این تمرین یک تصمیم هوشمندانه است که هم سرعت آموزش را حفظ می‌کند و هم توانایی مدل برای شناسایی ویژگی‌های مهم بیماری برگ را تضمین می‌کند. در صورت استفاده از سخت‌افزار قوی‌تر، می‌توان با کاهش اندازه‌ی پچ دقت مدل را بیشتر افزایش داد، هرچند باید هزینه‌ی پردازشی آن را هم در نظر گرفت.

۱-۴-۳- نمودار دقت و خطا و ماتریس آشفتگی مدل Vit

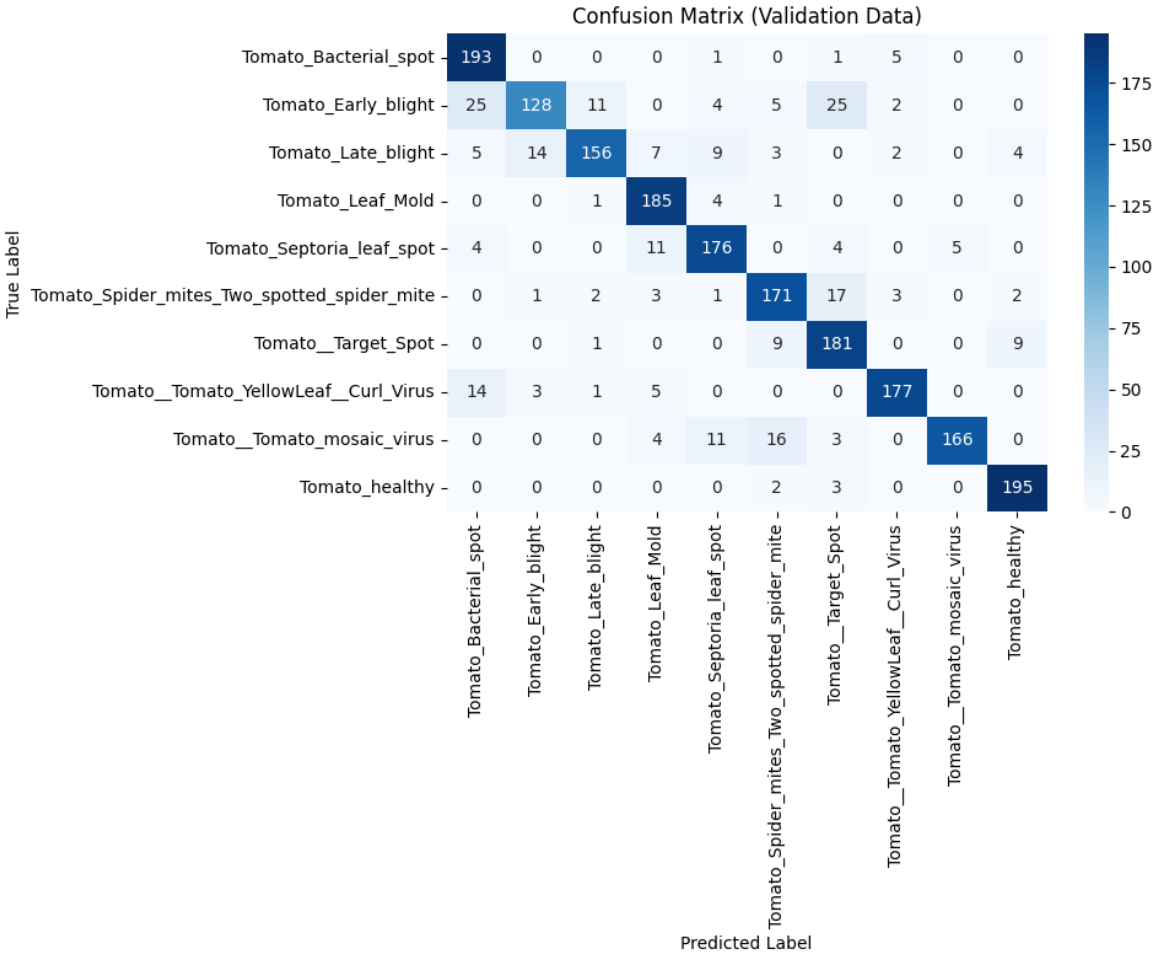


شکل ۳. نمودار دقت و خطا داده اعتبارسنجی و آموزش در Vit

نتایج به‌دست‌آمده از آموزش مدل Vision Transformer (ViT) برای شناسایی بیماری‌های برگ گوجه‌فرنگی، نمایانگر عملکرد قابل‌توجه و روند یادگیری باثبات مدل در طول ۳۰ دوره آموزشی است. نمودارهای دقت و خطا (...) نشان می‌دهند که مدل با شروعی نسبتاً ضعیف، در مدت کوتاهی توانسته است الگوهای تصویری بیماری‌ها را فراگرفته و بهبود چشم‌گیری در عملکرد خود ایجاد کند. کاهش یکنواخت و منظم در مقدار خطا برای هر دو مجموعه آموزش و اعتبارسنجی، در کنار افزایش تدریجی و پیوسته دقت، گویای آن است که مدل نه تنها در یادگیری داده‌های آموزشی موفق بوده، بلکه از تعمیم‌پذیری مناسبی روی داده‌های جدید برخوردار است.

دقت مدل از حدود ۳۰٪ در ابتدای آموزش به بیش از ۸۶٪ در دوره‌های پایانی رسیده است و در برخی نقاط، دقت روی داده‌های اعتبارسنجی از دقت آموزش نیز پیشی گرفته است. این مسئله معمولاً به دلیل

استفاده از تکنیک‌های افزایش داده (Data Augmentation) یا توزیع مناسب داده‌ها در آموزش رخ می‌دهد و به عنوان نشانه‌ای از یادگیری سالم و بدون بیش‌برازش (Overfitting) تلقی می‌شود.



شکل ۴. ماتریس آشفتگی مدل Vit

ماتریس آشفتگی (Confusion Matrix) نیز مکمل مناسبی برای درک رفتار مدل در سطح کلاس‌های جداگانه است. در این ماتریس، مشخص می‌شود که مدل در طبقه‌بندی بیماری‌هایی مانند لکه باکتریایی، ویروس موزائیک، و برگ‌های سالم عملکرد بسیار دقیقی داشته است. با این حال، در برخی بیماری‌ها نظیر پژمردگی زودرس و پیچش برگ زرد، مدل دچار درصدی از خطا و هم‌پوشانی شده است که با توجه به شباهت‌های بصری این بیماری‌ها، طبیعی و قابل انتظار است.

در مجموع، ViTتوانسته است با تکیه بر مکانیزم self-attention و تحلیل روابط global بین اجزای تصویر، ساختار پیچیده‌تری از ویژگی‌ها را نسبت به مدل‌های سنتی‌تر (مانندCNN) استخراج کند. دقت

نهایی مدل روی داده‌های اعتبارسنجی به حدود ۸۸٪ رسیده است، که برای یک معماری بدون لایه‌های کانولوشن و در کاربردی چالش‌برانگیز مانند طبقه‌بندی بیماری‌های گیاهی، کاملاً رضایت‌بخش است.

۱-۵- مقایسه و ارزیابی

۱-۵-۱- مقایسه مدل‌ها از نظر دقت، سرعت و پارامترها

جدول ۴. مقایسه دفت دو مدل CNN و Vit

مدل	دقت نهایی در اعتبارسنجی	روند همگرایی
CNN(inception-v3)	حدودا ۹۸٪	سریع، از ایپاک ۷ به بالا عملکرد عالی
Vit	حدودا ۸۸٪	روند آهسته‌تر ولی پیوسته تا ایپاک ۳۰

مدل CNN به‌وضوح دقت بالاتری در طبقه‌بندی نهایی ارائه داد. دقت در کلاس‌های مختلف بالا بود و confusion matrix نشان داد که اکثر نمونه‌ها به‌درستی تشخیص داده شدند. در مقابل، ViT اگرچه دقت قابل قبولی داشت، اما در برخی کلاس‌ها اشتباه بیشتری رخ داد.

جدول ۵. مقایسه سرعت دو مدل CNN و Vit

مدل	زمان آموزش ۳۰ ایپاک	علت تفاوت
CNN(inception-v3)	بسیار زیاد حدودا ۱ساعت و نیم	تصاویر بزرگ (۲۹۹×۲۹۹)، معماری سنگین با چند شاخه convolution
Vit	کم و سریع کمتر از ۱ ساعت	تصاویر کوچک‌تر(۶۴×۶۴)، attention ساده، معماری سبک‌تر

نکته قابل توجه :برخلاف انتظار رایج مبنی بر اینکه Transformer ها کندتر هستند، پیاده‌سازی سبک‌شده ViT با تصاویر کوچک‌تر، سرعت بسیار بیشتری نسبت به مدل CNN داشت. این تفاوت در عمل بسیار ملموس بود.

جدول ۶. مقایسه پارامترهای دو مدل CNN و Vit

مدل	تعداد تقریبی پارامترها	ساختار معماری
CNN(inception-v3)	زیاد، حدود ۲۳ میلیون	شبکه کانولوشنی با شاخه‌های Inception، لایه‌های عمیق
Vit	کم، حدود ۳–۲ میلیون	patch embedding، ۸ بلوک Transformer، MLP ساده

مدل ViT پیاده‌سازی‌شده در این پروژه دارای حدود ۲ تا ۳ میلیون پارامتر است، در حالی که مدل Inception V3 تقریباً ۲۳ میلیون پارامتر دارد. این اختلاف قابل توجه در تعداد پارامترها نشان‌دهنده‌ی سادگی، سبکی و بهینگی معماری ViT نسبت به مدل‌های کلاسیک CNN است. با وجود این سادگی، مدل ViT موفق شد به دقتی قابل قبول در حدود ٪۸۷ در داده‌های اعتبارسنجی دست یابد و در عین حال، زمان آموزش بسیار کوتاه‌تری را نسبت به CNN تجربه کرد (حدودا ۳۰ دقیقه در مقابل حدودا۹۰ دقیقه برای ۳۰ ایپاک).

در مقابل، مدل CNN با وجود پیچیدگی بسیار بیشتر، توانست دقت نهایی ٪۹۸ را ثبت کند و عملکرد دقیق‌تری در تفکیک کلاس‌های مختلف بیماری نشان دهد.

بررسی ماتریس درهم‌ریختگی (Confusion Matrix) نیز این اختلاف را تأیید می‌کند. مدل Inception V3 تقریباً تمام کلاس‌ها را با دقت بالا و خطای اندک تشخیص داد. در حالی که در مدل ViT، خطاهایی در تمایز بین کلاس‌هایی مانند Early Blight، Late Blight و YellowLeaf Curl Virusمشاهده شد. این موضوع نشان‌دهنده‌ی حساسیت بیشتر ViT به ویژگی‌های موضعی و جزییات تصویری است، که CNN با معماری کانولوشنی خود بهتر در استخراج آن‌ها عمل می‌کند.

نتیجه: مدل(Inception V3) CNN با وجود پیچیدگی بالا و تعداد زیاد پارامترها، بالاترین دقت و قابلیت اطمینان در طبقه‌بندی بیماری‌ها را ارائه می‌دهد و گزینه‌ای مناسب برای کاربردهای صنعتی و حساس است. در مقابل، مدل ViTسفارشی‌شده با معماری سبک‌تر، ابعاد ورودی کوچک‌تر (۶۴×۶۴)، و پارامترهای بسیار کمتر، توانست در زمانی بسیار کوتاه‌تر و با منابع محاسباتی محدودتر، عملکردی قابل قبول ارائه دهد. بنابراین، در سناریوهای با منابع محدود یا برای اهداف تحقیقاتی و توسعه سریع، ViT انتخابی مناسب است؛ در حالی که برای اهداف تجاری با نیاز به دقت بسیار بالا، CNN گزینه‌ی ارجح خواهد بود.

۱-۵-۲- در چه شرایطی مدل ضعیف‌تر میتوانست بهتر عمل کند؟

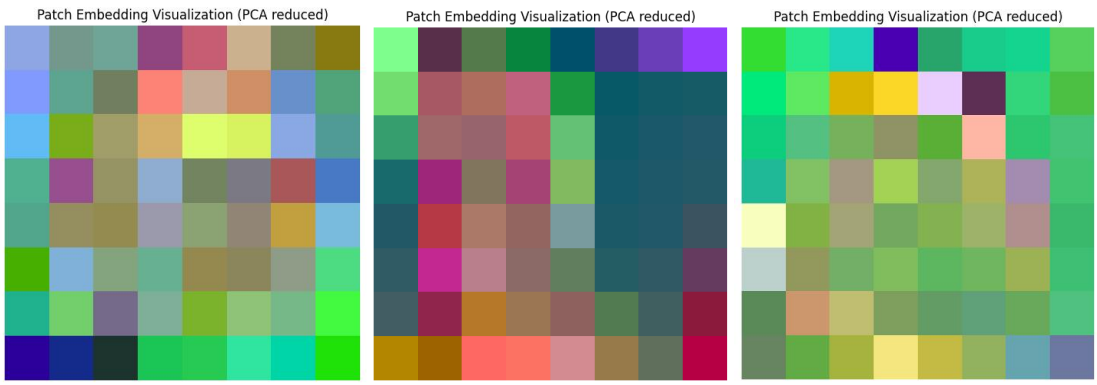
با وجود آن‌که مدل Inception V3 در این پروژه دقت بالاتری نسبت به مدل ViT ارائه داد، اما مدل ViTدر شرایط خاصی می‌توانست عملکرد بهتری داشته باشد. یکی از عوامل کلیدی، اندازه‌ی مجموعه داده است. مدل ViT معمولاً برای داده‌های بزرگ و متنوع طراحی شده است و در صورت دسترسی به مجموعه‌ای بزرگ‌تر از تصاویر بیماری‌های گوجه‌فرنگی، می‌توانست بهتر آموزش ببیند و از لحاظ دقت به سطح بالاتری برسد. دوم، در این پروژه ViT از پایه (بدونpretraining) آموزش داده شد؛ در حالی که استفاده از وزن‌های پیش‌آموزش‌دیده (pretrained weights) روی دیتاست‌هایی مشابه مانند ImageNet یا سایر پایگاه‌های داده کشاورزی، می‌توانست موجب بهبود چشم‌گیر عملکرد مدل شود.

علاوه بر این، مدل ViT با توجه جهانی (global self-attention) در تشخیص الگوهای کلی و وابستگی‌های دوربرد در تصویر بهتر از CNN عمل می‌کند. در مسائلی که تفاوت بین کلاس‌ها فقط در ویژگی‌های انتزاعی یا کلی تصویر است (و نه صرفاً در بافت یا ناحیه موضعی)، ViTمی‌تواند دقت بالاتری از CNN داشته باشد. همچنین، اگر تصاویر ورودی دارای نویز، چرخش، تغییرات روشنایی یا مقیاس متفاوت باشند، ViTبه‌دلیل ماهیتattention محور خود پایداری بیشتری نسبت به CNN خواهد داشت.

در نهایت، ViTبه‌دلیل ساختار ماژولار و انعطاف‌پذیر خود، گزینه‌ای مناسب برای توسعه آینده‌نگر محسوب می‌شود. در سناریوهایی که داده‌های چندحالته (Multi-modal) مانند ترکیب تصویر با متن، موقعیت مکانی یا اطلاعات سنسور مدنظر باشد، معماری ViT به‌مراتب قابلیت توسعه‌پذیری بیشتری نسبت به شبکه‌های CNN کلاسیک دارد.

۱-۶- امتیازی


```
Layer outputs shapes: {'cls_token_added': torch.Size([256, 65, 64]),
'pos_embedding_added': torch.Size([256, 65, 64]), 'transformer_blocks':
[torch.Size([65, 256, 64]), torch.Size([65, 256, 64]), torch.Size([65, 256,
64]), torch.Size([65, 256, 64]), torch.Size([65, 256, 64]), torch.Size([65,
256, 64]), torch.Size([65, 256, 64]), torch.Size([65, 256, 64])], 'norm':
torch.Size([65, 256, 64]), 'head_output': torch.Size([256, 10])}
```

شکل ۵. نمایش خروجی لایه Patch Embedding به صورت تصویر

برای این بخش، در طول آموزش مدلViT ، خروجی لایه‌ی Patch Embedding برای یک تصویر نمونه در ایپاک اول استخراج و به‌صورت بصری نمایش داده شد. در این مرحله، تصویر ۶۴×۶۴ ورودی به ۸×۸ پچ تقسیم شده و هر پچ توسط لایه‌ی Patch Embedding به یک بردار ۶۴ بعدی تعبیه شد. سپس با استفاده از روش PCA، این بردارهای ۶۴ بعدی به ۳ بعد اصلی کاهش یافته و به‌عنوان تصویر RGB با ابعاد ۸×۸ قابل نمایش شدند.

خروجی این تابع، تصویری رنگی از بردارهای embedding پچ‌های تصویر ورودی است. رنگ‌های متفاوت بیانگر تفاوت در اطلاعات هر پچ هستند. این نمایش کمک می‌کند تا درک بهتری از نحوه‌ی پردازش تصویر توسط مدل در مراحل ابتدایی به‌دست آید و نشان دهد که ViT چگونه اطلاعات موضعی را به بردارهای قابل استفاده در ساختار attention تبدیل می‌کند.