تمرین دوم پایتون
شبکه‌های عصبی بهار ۱۴۰۲

آریا ابراهیمی ۹۸۲۲۷۶۲۱۷۵

هدف از این تمرین، بررسی انتقال دانش در شبکه‌های عصبی عمیق می‌باشد. برای این کار از سه شبکه VGG16، ResNet50 و DenseNet121 آموزش داده شده بر روی ImageNet استفاده می‌کنیم. شبکه های یادگیری شده در لایه ابتدایی شامل فیچر های ابتدایی تر می‌شوند و هرچقدر عمیق‌تر می‌شوند، ویژگی های پیچیده‌تری را شامل می‌شوند. هدف این است تا با استفاده از وزن های یادگیری شده لایه های ابتدایی با تعداد داده کمتر و سرعت بیشتری بتوانیم classification را انجام دهیم.

در این تمرین با استفاده از داده های سگ و گربه، عمل‌کرد شبکه های معرفی شده را با درصد های متفاوت لایه های freeze شده بررسی می‌کنیم. در نهایت ویژگی های یادگیری شده مربوط به داده‌ها را به یک RandomForestClassifier میدهیم و تسک نهایی مربوط به classification را با استفاده از آن نیز انجام می‌دهیم و با classification انجام شده توسط تک نورون لایه آخر شبکه مقایسه می‌کنیم.

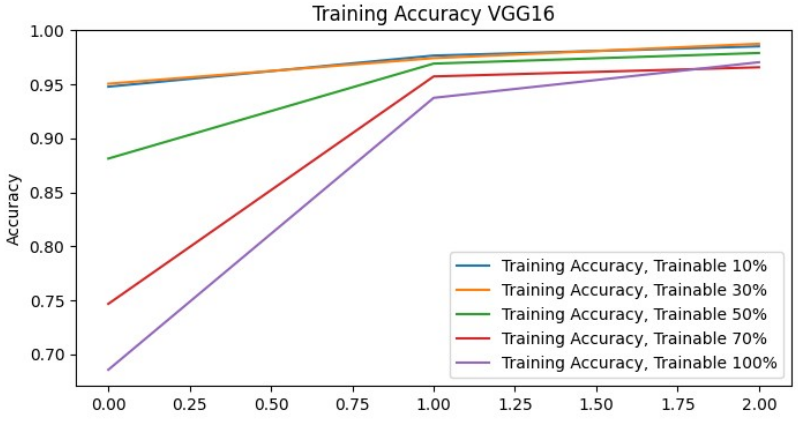
۱- بخش اول

همان‌طور که در جدول ۱ مشاهده می‌شود، هرچقدر تعداد لایه های trainable بیشتر شود، میزان accuracy مدل کمتر می‌شود و بهترین عمل‌کرد مربوط به تعداد لایه های قابل یادگیری ۱۰ درصد و ۳۰ درصد می‌باشد و این برای هر سه شبکه VGG16 و ResNet50 و DenseNet121 برقرار است. می‌توان نتیجه گرفت که انتقال دانش می‌تواند به یادگیری بهتر کمک کند و اگر داده زیادی برای آموزش یک شبکه نداریم، استفاده از یک شبکه از پیش آموزش داده شده می‌تواند از overfit شدن و مشکلات دیگر جلوگیری کند.

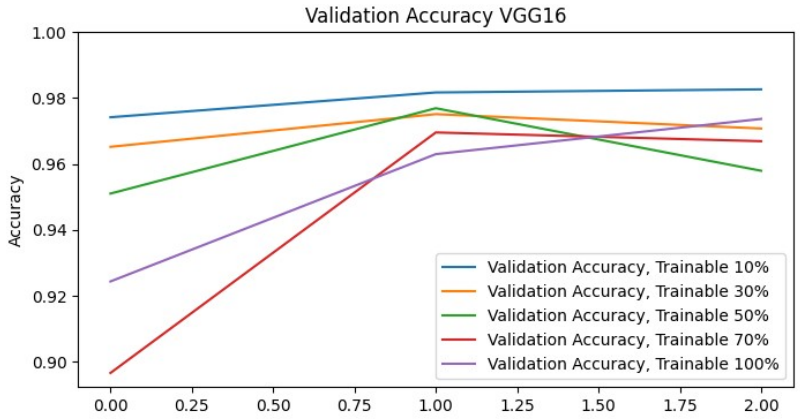
جدول ۱:دقت شبکه های انتقال دانش یافته از مدل های گفته شده بعد از سه epoch بر روی دیتاست سگ و گربه.

Model	VGG16					ResNet50					DenseNet121				
Trainable	10%	30%	50%	70%	100%	10%	30%	50%	70%	100%	10%	30%	50%	70%	100%
Train Accuracy	98.51	98.75	97.9	96.57	97.04	99.57	99.64	99.51	99.08	99.23	99.46	99.49	99.48	99.44	99.26
Validation Accuracy	98.27	97.08	95.80	96.69	97.37	98.75	98.68	98.66	98.22	97.63	99.13	99.15	99.03	97.99	98.83

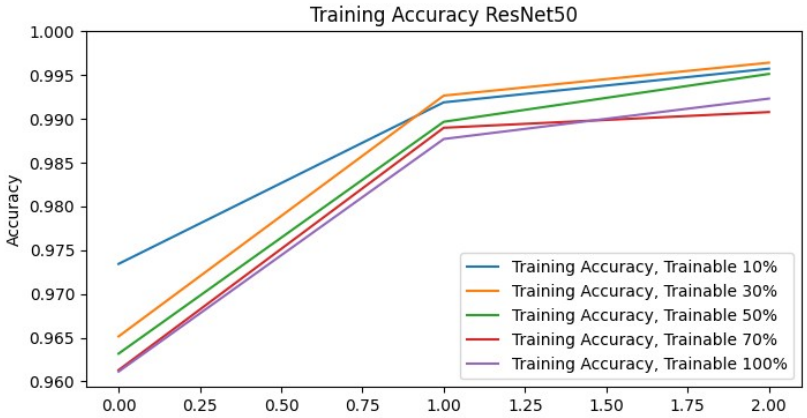
همان‌طور که مشاهده می‌شود در مجموع DenseNet121 عملکرد بهتری نسبت به دو مدل دیگر دارد و در هر مدل هم لایه های قابل یادگیری بین ۱۰ درصد تا ۳۰ درصد بهترین نتیجه را نمایش می‌دهند و همچنین نسبت به کانفیگ های دیگر سریعتر یاد می‌گیرند.



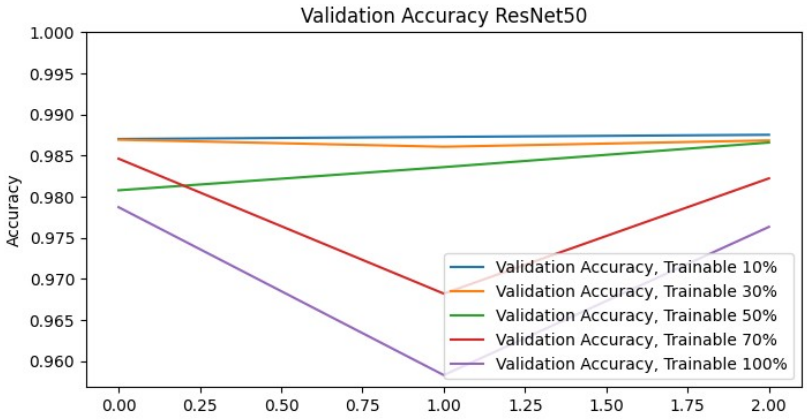
شکل ۱: دقت Training برای شبکه های انتقال دانش یافته از VGG16



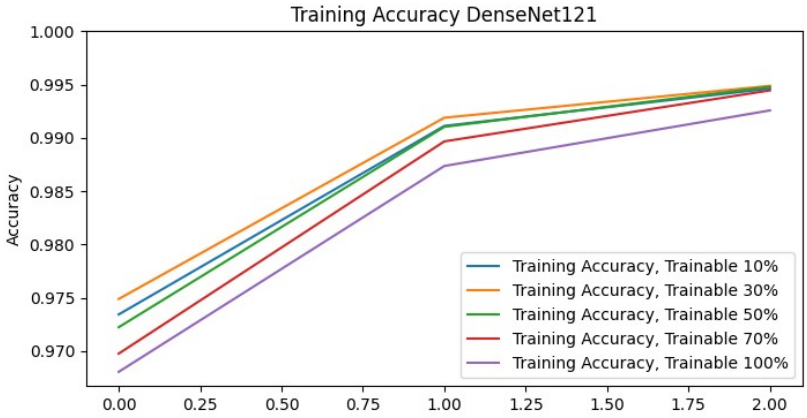
شکل ۲: دقت Validation برای شبکه های انتقال دانش یافته از VGG16



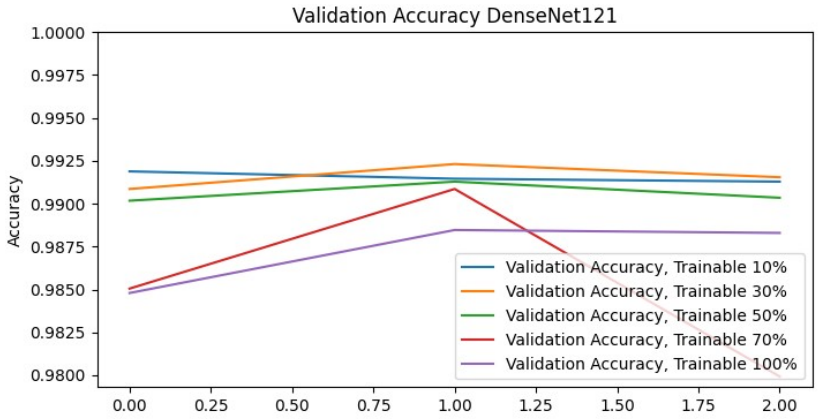
شکل ۳: دقت Training برای شبکه های انتقال دانش یافته از ResNet50



شکل ۴: دقت Validation برای شبکه های انتقال دانش یافته از ResNet50



شکل ۵: دقت Training برای شبکه های انتقال دانش یافته از DenseNet121



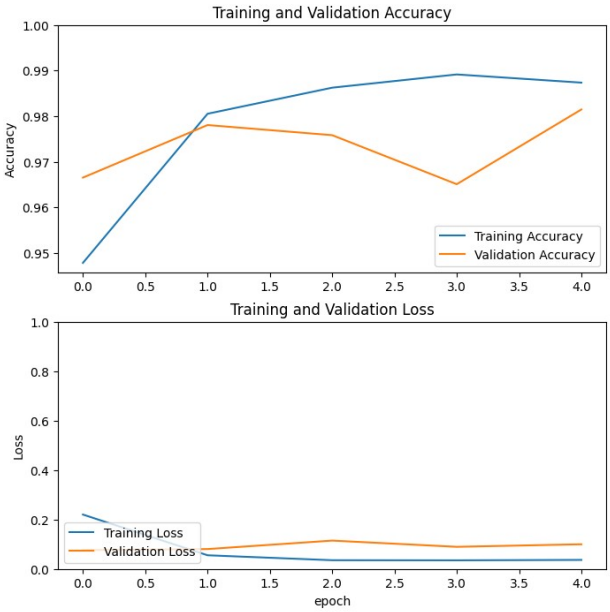
شکل ۶: شکل ۵: دقت Validation برای شبکه های انتقال دانش یافته از DenseNet121

سوالی که پیش می‌آید این است که آیا اگر تمامی لایه های مدل اصلی را freeze کنیم نسبت به نتایج فعلی خروجی بهتری خواهیم داشت؟ نتایج به دست آمده برای زمانی که تمامی لایه ها را freeze کنیم در جدول۲ قابل مشاهده است.

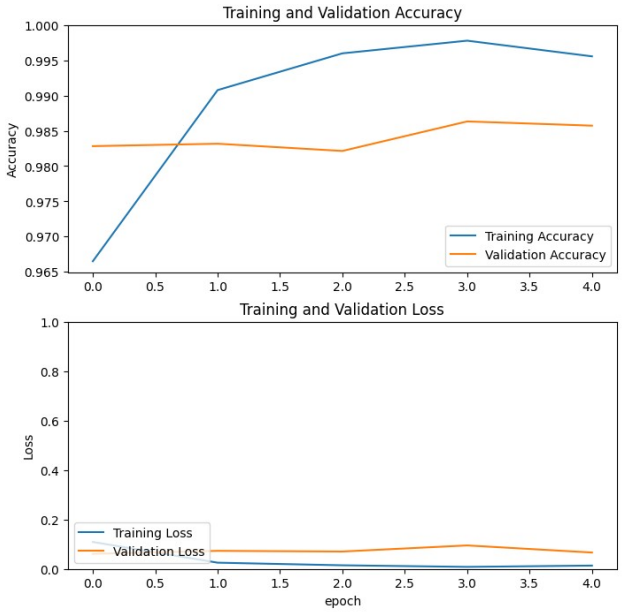
جدول ۲: انتقال دانش بعد از سه epoch زمانی که تمامی لایه ها freeze شوند

Model	VGG16	ResNet50	DenseNet121
Train Accuracy	98.46	99.17	99.26
Validation Accuracy	97.18	98.78	98.99

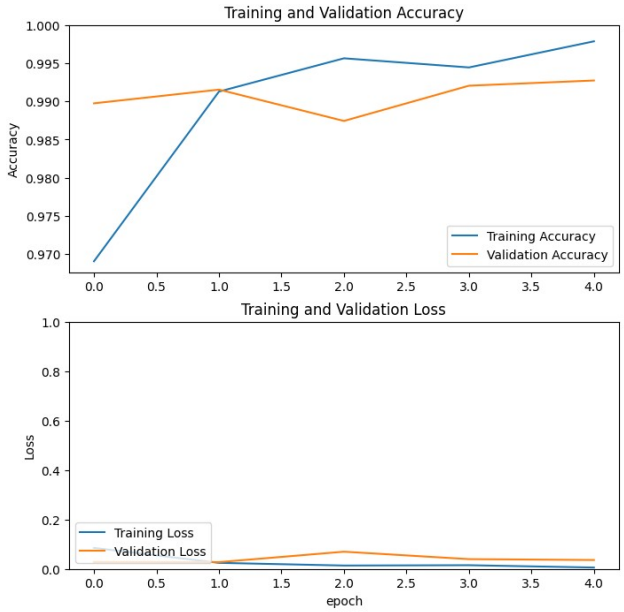
همان‌طور که مشاهده می‌شود در حالتی که تمامی لایه های شبکه های گفته شده freeze شوند، نتیجه بهتری مشاهده نمی‌شود.



شکل ۷: انتقال دانش با استفاده از مدل VGG16 و ۳۰ درصد لایه های trainable در پنج epoch



شکل ۸: انتقال دانش با استفاده از مدل ResNet50 و ۳۰ درصد لایه های trainable در پنج epoch



شکل ۹: انتقال دانش با استفاده از مدل DenseNet121 و ۳۰ درصد لایه های trainable در پنج epoch

۲- بخش دوم

در این بخش، ویژگی های یادگیری شده مربوط به نسخه ۳۰ درصد شبکه ها را برای همان دیتاست استخراج کرده و با استفاده RandomForestClassifier کتابخانه SciKit آن‌ها را دسته‌بندی می‌کنیم. چالشی که در قسمت پیاده سازی این بخش وجود دارد این است که دیتاست را به صورت BatchDataset دریافت کرده ایم (شامل داده های یادگیری و لیبل مربوط به آنها) که باعث می‌شود به راحتی به داده و لیبل دسترسی نداشته باشیم و از طریق متد take میتوانیم روی batch ها پیمایش کنیم. یک راه میتواند این باشد که همزمان با پیمایش روی داده‌ها، آن‌ها را به همراه لیبل متناظرشان ذخیره کنیم. این روش رم زیادی اشغال می‌کند و باعث کرش کردن محیط Google Colab می‌شود.

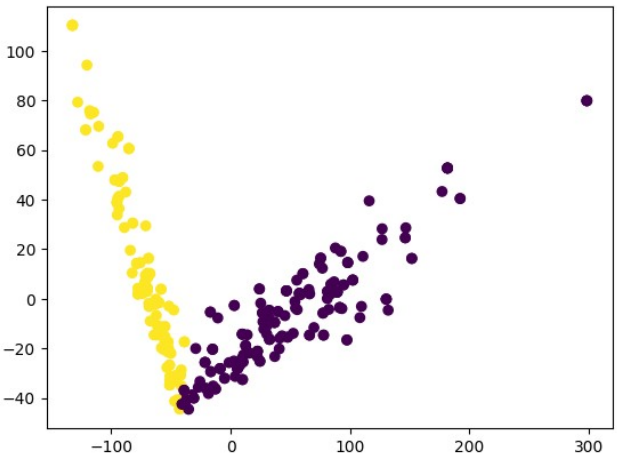
راه دیگر این است که RandomForestClassifier را همزمان با پیمایش روی دیتاست به صورت mini-batch آموزش دهیم. با این کار دیگر نیازی به ذخیره داده ها نیست ولی زمان بیشتری نیاز دارد.

دقت RandomForestClassifier برای داده های validation برای سه شبکه در نسخه ۳۰ درصد در جدول ۳ قابل مشاهده است.

جدول ۳: دقت بر روی دیتاست validation با استفاده از RandomForestClassifier

Model	VGG16	ResNet50	DenseNet121
Validation Accuracy	97.12	98.74	99.25

همان‌طور که مشاهده می‌شود، طبقه بندی با استفاده از RandomForestClassifier روی ویژگی های استخراج شده از شبکه ها نتایج برابر و حتی در مواردی بهتر از classification خود شبکه ها ارائه می‌دهد.



شکل ۱۰: ویژگی های کاهش dimensionality داده شده توسط PCA و یادگیری شده توسط شبکه (قبل از اعمال تابع فعال‌ساز relu)

می‌توان نتیجه گرفت که مهم‌ترین کاری که شبکه ها انجام میدهند این است که ویژگی یاد می‌گیرند و شاید تسک classification را طبقه‌بندی مثل RandomForestClassifier بهتر انجام دهد ولی چیزی که شبکه های عصبی را متمایز می‌کند و باعث می‌شود خوب عمل کنند، ویژگی های یادگیری شده خوب است.