گزارش پیاده‌سازی و پاسخ به سوالات

سوال الف) (سوال تئوری) در اتصال باقی‌ماندگی دو حالت کلی وجود دارد که می‌تواند اتصالات کوتاه یا بلند باشند. برای تعیین نوع اتصالات می‌تون موارد زیر را مد نظر قرار داد:

- اتصالات کوتاه معمولا بر روی لایه‌های پیچشی متوالی اعمال می‌شود و ابعاد ورودی در آن‌ها تغییر نمی‌کند یا تغییرات کمی دارد که مانند آن را در شبکه ResNet می‌بینیم.
- در معماری کدگذار-کدگشا معمولا در قسمت کدگذار ابعاد ورودی تغییر زیادی می‌کند و در قسمت کدگشا به حالت قبل برمی‌گردد که در این حالت از اتصالات باقی‌ماندگی طولانی استفاده می‌شود که قسمت‌های هم بعد در کدگذار و کدگشا را به هم وصل می‌کند.
- استفاده از اتصالات بلند باعث می‌شود اطلاعات محلی آن قسمت به قسمت‌های دور انتشار یابد و چون هر قسمت از شبکه‌هایِ شامل لایه‌های پیچشی معمولا ویژگی‌های خاصی از تصاویر را استخراج می‌کنند، این کار باعث می‌شود ویژگی‌های به دست آمده در قسمت‌های مختلف که به دلیل فاصله زیاد تفاوت بیشتری هم با هم دارد، با یکدیگر ترکیب شوند و مدل به صورت کلی‌تری انواع ویژگی‌های تصاویر را در نظر بگیرد که در کاربرد خاص ممکن است مورد استفاده باشد.
- به طور کلی این موضوع وجود دارد که اطلاعات کلی‌تر بیشتر به پاسخ «چگونه؟» می‌پردازند و اطلاعات محلی‌تر به پاسخ «کجا؟». این موضوع نشان می‌دهد که اتصالات بلند به حل مسائلی که نیاز به فهم چگونگی تصاویر دارند کمک می‌کند در حالی که اتصالات کوتاه به حل مسائلی که نیاز به فهم محل اطلاعات جزئی تصویر دارند کمک می‌رساند که بسته به کاربرد مورد نظر یکی از آن‌ها بیشتر مورد توجه ما خواهد بود.

سوال ب) (سوال تئوری) در شبکه‌های عصبی پیچشی متراکم چندین Dense Block داریم که با لایه‌های انتقال به ترتیب به هم وصل شده‌اند. در هر یک از این لایه‌های انتقال ابتدا یک لایه کانولوشن ۱*۱ داریم و سپس از average polling استفاده شده است. در واقع در لایه انتقال، چندین ورودی که concat شده‌اند ابتدا به لایه کانولوشن ۱*۱ می‌روند و کاهش بعد می‌یابند تا جمع وزن‌داری از تمام ورودی‌ها در هر کانال خروجی وجود داشته باشد و سپس از یک average polling بر روی آن استفاده می‌شود. دلیل استفاده از average polling آن است که بتوان ترکیبی از اطلاعات تمام قسمت‌های داخلی Dense Block قبلی را در خروجی داشت و این نوع polling با استفاده از عبور گرادیان بر روی تمام ایندکس‌های متناظر، بهتر می‌تواند ویژگی کلی ورودی‌ها را نمایش داده و انتقال دهد که با مفهوم اصلی شبکه DensNet که اتصال بین هر دو لایه را در نظر دارد تطابق

بیشتری دارد. در حالی که استفاده از max polling گرادیان را تنها بر روی ایندکس مقدار max عبور می‌دهد و تمام اطلاعات قبلی را منتقل نمی‌کند بلکه سعی می‌کند تنها قوی‌ترین سیگنال ورودی را در نظر بگیرد.

سوال ج) (سوال تئوری) برای طراحی یک مدل عصبی پیچشی که شامل استخراج ویژگی و دسته‌بندی است ابرپارامترهای زیادی وجود دارند که باید مقدار مناسب برای آن‌ها تعیین شود. یکی از روش‌هایی که به نظر منطقی می‌رسد آن است که برای قسمت دسته‌بندی که لایه‌های انتهایی شبکه هستند چند لایه خطی با تعداد نورون‌ها و تعداد لایه‌های دلخواه قرار دهیم و ابرپارامترهای قسمت استخراج ویژگی را تغییر دهیم و هر بار با آموزش مدل تا چند ایپاک و ارزیابی آن سعی کنیم به پارامترهای بهینه در قسمت استخراج ویژگی برسیم. برای این قسمت می‌توان هر بار پارامترهای یک لایه پیچشی را تغییر داد و سایر پارامترها را ثابت در نظر گرفت تا به تدریج مقادیر بهینه برای هر لایه به دست آید. در مرحله بعد ابرپارامترهای به دست آمده برای قسمت استخراج ویژگی را ثابت فرض می‌کنیم و تعداد لایه‌ها و نورون‌ها را در قسمت دسته‌بندی تغییر می‌دهیم تا به مقادیر بهینه برای آن‌ها برسیم.

دسته بندی تصاویر:

۱) (پیاده‌سازی) داده‌های مورد نظر دانلود شده است و ابتدا یک پیش‌پردازش بر روی آن‌ها انجام می‌شود. در قدم اول ۱۰ درصد از داده‌های آموزشی برای ارزیابی جدا شده‌اند و داده‌های آزمون هم که به صورت مجزا وجود دارند. سپس میانگین و انحراف معیار مقادیر هر یک از سه کانال RGB برای همه تصاویر آموزشی به دست می‌آید که به صورت زیر هستند:

```
images mean:  [0.26888581 0.25730865 0.27832698]
images std:  [0.26888581 0.25730865 0.27832698]
```

سپس هر یک از تصاویر با استفاده از میانگین و انحراف معیار نوشته شده نرمال می‌شود. در ادامه کلاس Dataset پیاده‌سازی شده که داده‌ها را بارگذاری کرده و برمی‌گرداند و از Dataloader با batch size برابر ۶۴ استفاده شده است.

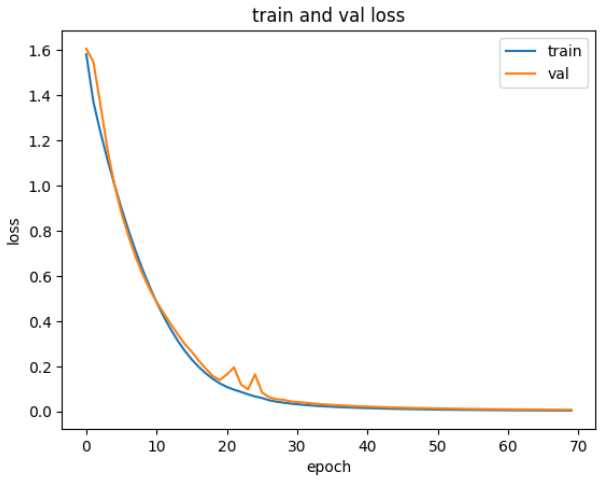
شبکه طراحی شده با آزمون و خطای بسیار به دست آمده است که در آن از ۵ لایه پیچشی استفاده شده است که اطلاعات این لایه ها به صورت زیر است. همچنین بعد از هر لایه پیچشی از یک Batch Normalization و یک لایه Max Polling استفاده شده که اطلاعات آن‌ها را در جدول زیر می بینیم:

Batch Normalization	Max polling		Convolution					
	stride	Kernel size	passing	stride	Kernel size	Output channel	Input channel	
16	2	3	2	1	5*5	16	3	۱
64	2	3	2	1	5*5	64	16	۲
96	2	2	2	1	5*5	96	64	۳
128	2	2	1	1	3*3	128	96	۴
128	2	2	1	1	3*3	128	128	۵

همچنین در انتهای مدل دو لایه خطی قرار گرفته‌اند که اولی ابعاد را از ۱۲۸*۴*۴ به ۱۲۸ و دومی از ۱۲۸ به ۵ انتقال می‌دهد.

برای تابع خطای مدل از cross entropy و برای تابع بهینه‌ساز از Adam با نرخ یادگیری 5e-5 استفاده شده است. این مقادیر در تمام قسمت‌های بعدی ثابت باقی مانده‌اند.

پس از آموزش مدل تا ۷۰ ایپاک تغییرات loss بر روی داده آموزشی و ارزیابی به صورت زیر است:

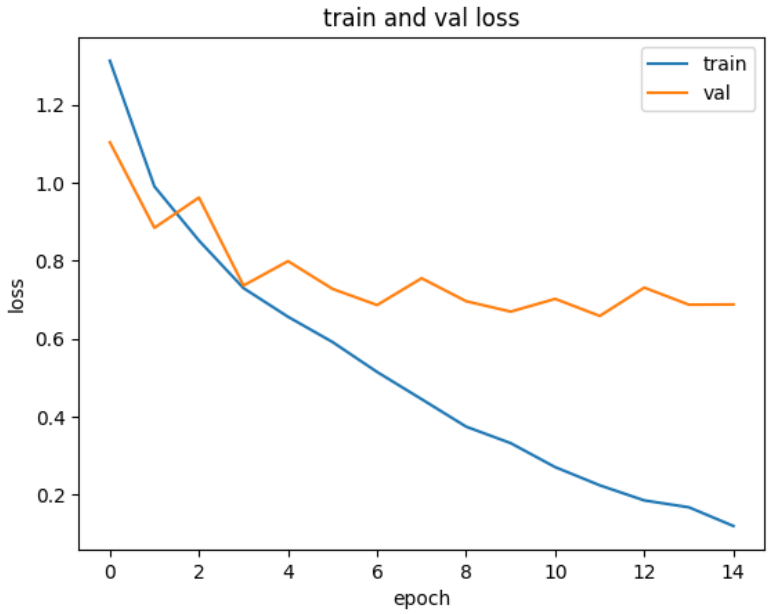


سپس داده‌های آزمون را استفاده می‌کنیم و مقادیر Precision و recall و F1 را به دست می آوریم که نتایج به صورت زیر است:

F1	recall	Precision	

0.614	0.616	0.622	Test data

۲) افزودن اتصال باقی‌مانده (پیاده‌سازی، امتیازی): در شبکه توضیح داده شده در قسمت قبل اتصال باقی‌ماندگی ایجاد می‌کنیم. نحوه اجرای این کار آن است اتصالات مختلف را بررسی کرده‌ایم و در انتها بهترین اتصال به این صورت است که بین لایه پیچشی دوم و آخری یک اتصال برقرار باشد. این اتصال با استفاده از یک لایه پیچشی ۵*۵ با stride برابر ۲ انجام شده است. تغییرات نمودار loss برای داده آموزشی و ارزیابی در طی آموزش به صورت زیر است:

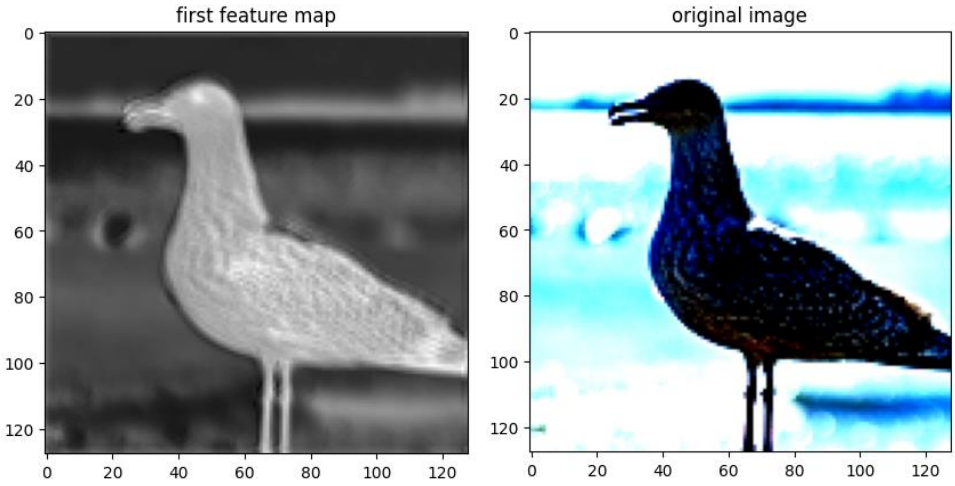


معیارهای به دست آمده از داده آزمون نیز به صورت جدول زیر می‌باشند.

F1	recall	Precision	
0.749	0.747	0.762	Test data

همان‌طور که دیده می‌شود در اثر اعمال لایه باقی‌ماندگی معیارهای به دست آمده بر روی داده آزمون نتایج بهتری دارد که دلیل آن افزودن یک اتصال از لایه‌های ابتدایی به انتهایی است که باعث شده به آموزش مدل کمک کند و دقت و صحت پیش‌بینی را افزایش دهد. علاوه بر آن با بررسی نمودار تغییرات خطا دیده می‌شو که مدل قبلی بعد از حدود ۷۰ ایپاک به همگرایی رسید ولی در مدل فعلی تنها بعد از ۱۵ ایپاک نمودار خطای ارزیابی به همگرایی رسیده است که نشان از آن دارد که استفاده از لایه باقی‌مانده در مدل باعث بهبود سرعت همگرایی مدل نیز شده است.

۳) نمایش ماتریس ویژگی(پیاده‌سازی): یکی از تصاویر موجود در داده‌های آموزشی را جدا کرده‌ایم و آن را به مدل آموزش دیده در قسمت ۱ می‌دهیم و یکی از خروجی‌های لایه پیچشی اول را جدا می‌کنیم و آن را نمایش می‌دهیم. تصاویر زیر شکل اصلی و خروجی به دست آمده را نشان می‌دهد:

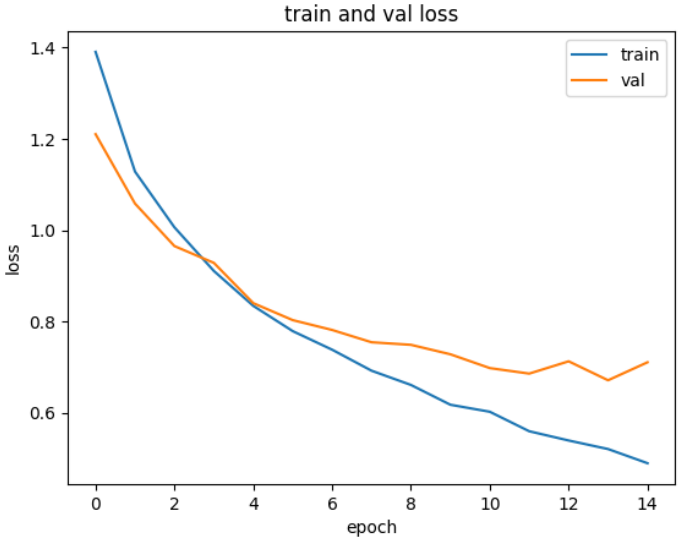


همان‌طور که دیده می‌شود در خروجی این لایه پیچشی لبه‌های تصویر به خوبی تشخیص داده‌شده و با رتگ تیره‌تر در تصویر دیده می‌شود.

۴) افزودن بلوک `inception` (پیاده‌سازی): بلوک گفته شده در سوال طراحی شده است که تعداد کانال‌های ورودی و خروجی آن با بررسی متعدد به صورت جدول زیر انتخاب شده‌اند. سپس این بلوک به جای هر یک از لایه‌های پیچشی مدل ساخته شده در قسمت ۱ قرار گرفته است و بین هر دو بلوک از max polling استفاده شده است که ابرپارامترهای این polling ها به صورت گفته شده در جدول زیر هستند. لایه‌های خطی نهایی هم بدون تغییر باقی‌مانده‌اند.

Max polling		Output channels	Input channels	
stride	Kernel size			
2	3	64	3	Block 1
2	3	96	64	Block 2
2	2	96	96	Block 3
2	2	128	96	Block 4
2	2	128	128	Block 5

سپس مدل بر روی داده‌های آموزشی تا ۱۵ ایپاک آموزش دیده است که تغییرات loss آن برای داده‌های آموزشی و ارزیابی به صورت زیر است:



در انتها بر روی داده آزمون مدل ارزیابی می‌شود و معیارهای زیر به دست می‌آیند:

F1	recall	Precision	
0.730	0.731	0.738	Test data

با مقایسه نتایج به دست آمده از این مدل با مدل قسمت اول دیده می‌شود که دقت و صحت پیش‌بینی‌ها افزایش چشم‌گیری داشته است که در اثر استفاده از بلاک‌های inception می‌باشد. علاوه بر بهبود دقت، همان‌طور که از نمودار تغییرات خطا مشخص است، آموزش مدل قسمت ۱ نیاز به ۷۰ ایپاک آموزش داشت ولی مدل فعلی تنها در ۱۵ ایپاک به همگرایی می‌رسد. البته به دلیل لایه‌های پیچشی بیشتر تعداد پارامترهای مدل بیشتر شده و زمان آموزش در هر ایپاک از مدل فعلی تقریبا ۶ برابر مدل قبلی است.

انتقال یادگیری:

۵) (سوال تئوری) در گام اول انتقال یادگیری قسمت استخراج ویژگی غیرقابل آموزش تنظیم می‌شود تا در حین این آموزش وزن‌های این قسمت تغییری نکنند و در نتیجه خروجی استخراج ویژگی به ازای ورودی یکسان در طی آموزش مقادیر یکسانی باشد. در این حالت لایه دسته‌بندی که بر روی مدل قرار گرفته، به ازای ورودی داده یکسان، ویژگی‌های یکسانی از مدل استخراج ویژگی دریافت می‌کند و می‌تواند وزن‌هایش را به گونه‌ای تنظیم کند

که از این ویژگی‌ها به برچسب‌های مورد نظر برسد. در حالی که اگر قسمت استخراج ویژگی هم آموزش ببیند، خروجی حاصل از آن در طی آموزش تغییر می‌کند و آموزش دسته‌بند به خوبی انجام نخواهد شد.

۶) انتقال یادگیری (پیاده‌سازی): در این قسمت ابتدا داده‌های مورد نظر را دانلود کرده و پردازش کرده‌ایم. در مرحله اول هر یک از داده‌ها به ابعاد ۱۲۸*۱۲۸ تغییر اندازه داده شده‌اند. سپس مانند قسمت قبل میانگین و انحراف معیار هر یک از کانال‌ها RGB در داده‌ها به دست آمده و بر اساس آن‌ها داده‌ها نرمال‌سازی شده‌اند که این مقادیر به صورت زیر است:

```
images mean: [0.29280583 0.26505889 0.28605474]
images std: [0.29280583 0.26505889 0.28605474]
```

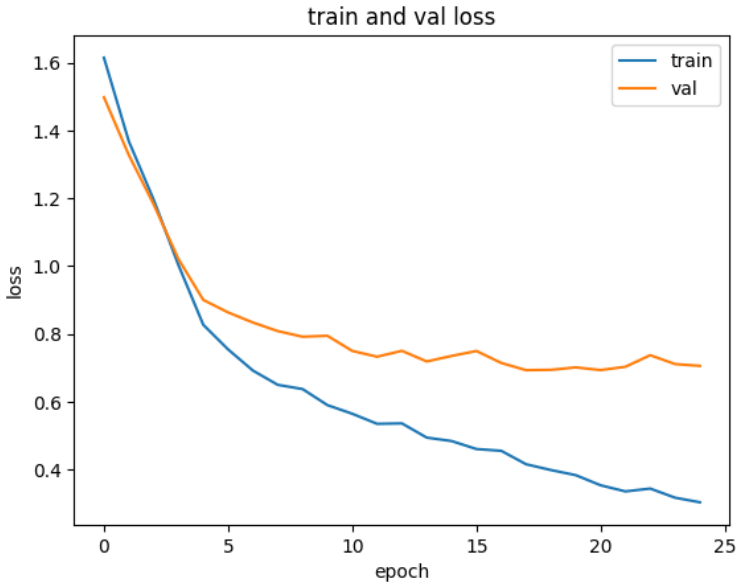
سپس مدلی طراحی شده است که مدل آموزش دیده در قسمت قبل را می‌گیرد و یک دسته‌بند بر روی آن قرار می‌دهد. دسته‌بند اضافه شده شامل دو لایه خطی با تعداد نورون‌های ۱۲۸ و ۵ است که ۵ در واقع تعداد کلاس‌های داده‌های جدید است.

همچنین تابعی در کلاس مدل مورد نظر نوشته شده است که لایه‌های قابل آموزش را تعیین می‌کند و سایر لایه‌ها را غیر قابل آموزش می‌کند.

توضیح سوال پرسیده شده: با توجه به آن که داده‌هایی که مدل از پیش آموزش دیده بسیار متنوع‌تر هستند و تنها یکی از کلاس‌های آن گل‌ها هستند آموزش آن و نحوه استخراج ویژگی در آن متفاوت است. با این وجود مدل خواهد توانست ویژگی‌های اصلی تصاویر را استخراج کند ولی باید پس از آموزش اولیه‌ی لایه‌های دسته‌بند جدید، به تعداد گام مناسبی لایه‌های مدل استخراج ویژگی نیز آموزش ببینند تا بتواند ویژگی‌های متناسب با داده‌های جدید را به دست آورد.

نحوه آموزش مدل به این صورت است که در ایپاک اول تنها لایه‌های دسته‌بندی که بر روی مدل از پیش آموزش دیده قرار گرفته بود آموزش می‌بینند. سپس در ۶ ایپاک بعدی به ترتیب ابتدا لایه‌های خطی انتهای شبکه پیش‌آموزش دیده و سپس از انتها به ابتدا بلاک‌های inception را قابل آموزش می‌کنیم. سپس در بقیه ایپاک‌های آموزش تا ۲۵ ایپاک کل مدل قابل آموزش می‌باشد.

نمودار تغییرات خطای آموزشی و ارزیابی در مراحل آموزش به صورت زیر است که کاهش خطا در مراحل آموزش به خوبی در آن دیده می‌شود:



سپس بر روی داده‌های آزمون مدل را ارزیابی می‌کنیم و معیارهای مورد نظر به صورت زیر یافته می‌شوند:

F1	recall	Precision	
0.704	0.703	0.725	Flower Test data

دیده می‌شود که با وجود آن که مدل از پیش آموزش دیده بر روی مجموعه داده متفاوتی با دسته‌بندی متفاوتی آموزش دیده بود با استفاده از فرایند گفته شده از این مدل آموزش‌دیده در دسته‌بندی داده‌های جدید استفاده شده است و در نهایت مدل توانسته به دقت و صحت معقولی در دسته‌بندی داده‌های جدید برسد.