گزارش تمرین دوم

مهرانه مقتدائی فر 97222086

سوال اول:

در ابتدا برای پردازش داده ها مشاهده میکنیم که 3 فایل متفاوت برای test,train,valid داریم و یک فایل csv که در آن ID هر تابلو راهنمایی نمایش داده شده است. تعداد این کلاس های متفاوت 43 تا میباشد. یک تصویر از هر کلاس را نمایش دادم و با توجه به تصاویر مشخص است که شرایط ظاهری آن ها یکسان نیست و حتی برخی بسیار تاریک و غیر قابل مشاهده هستند.
با مشاهده داده های train مشخص است که آرایه ای از عکس ها به ما داده شده که ویژگی آنها در features است ما این بخش را انتخاب کرده و به عنوان X_train در نظر میگیریم و همچنین labels را نیز به عنوان y_label در نظر گرفتیم
سپس با رسم نموداری نشان دادیم که فراوانی کلاس های متفاوت به چه صورت است. با توجه به نمودار مشخص است که کلاس های ما متوازن نیستند.

در ابتدا شبکه را بدون اعمال هیچ تغییری بر روی عکس ها یک بار کامل train میکنیم.
با استفاده از image data generator یک تغییراتی در عکس ها به صورت رندم اعمال میشود که این باعث میشود تا شبکه عکس ها را در حالت های مختلف بررسی کند و نتیجه نهایی آن بهتر باشد.
برای پردازش تصویر ها باید از CNN استفاده کنیم. شبکه ای که دارای لایه های convolution است.
همچنین عکس های ما 3 کانال رنگی دارند.

با توجه به مدل مشخص است که لایه های conv برای بررسی قسمت های مختلف عکس ها قرار داده شده است(خروجی این لایه ها را در انتها برای یک عکس خاص نشان میدهیم)
لایه batch normalization نیز اعمال شده تا داده های ورودی را normalize کند و تغییراتی بر روی آنها اعمال میکند تا همگرایی شبکه سریع تر شود.

با train کردن شبکه بدون اعمال هیچ تغییری بر روی عکس ها نتیجه نسبتا خوبی داریم اما بهتر است که تغییر های مختلف برروی عکس ها اعمال کنیم و تا حد امکان شرایط آنها را یکسان کنیم تا نتیجه بهتری(مخصوصا برروی داده های تست) بگیریم.

پاسخ به سوال های داده شده :

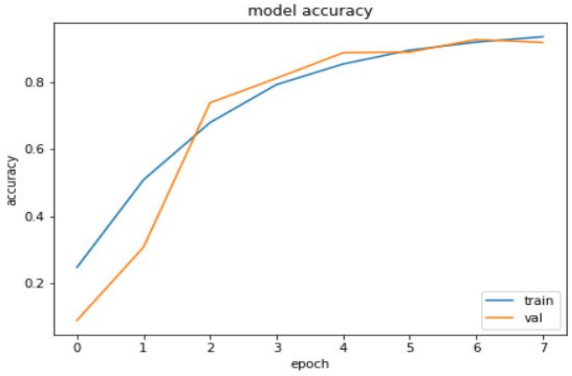
- با توجه به نموداری که در ابتدا بیان شد، مشخص است که کلاس های ما متوازن نیستند، با کدی که بعد از آن زده شده نیز میتوان متوجه شد که به طور دقیق از هر کلاس چه تعداد وجود دارد، مینیمم آنها 180 و ماکزیمم آنها 2010 است. برای متوازن کردن این کلاس ها میتوانیم از روش های مختلفی مانند upsampling و یا downsampling و یا حتی از وزن دادن به داده ها استفاده کنیم. Upsampling به این معنی که عکس های کلاسی که فراوانی کمتری دارد زیاد کرده تا به تعداد ماکزیمم برسد و downsampling به این معنا است که از کلاس هایی که تعداد داده های زیادی دارند به طور رندم داده حذف کنیم تا به نزدیک مینیمم برسیم. هر کدام از این روش ها را امتحان کردم اما برای این دسته داده ها تاثیر زیادی نداشت.

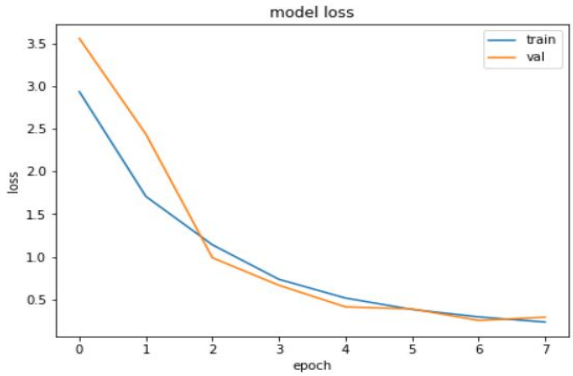
- عکس های ما 3 کانال رنگی دارند و برای غیر رنگی کردن آنها از دستوری در cv2 استفاده شده که کانال رنگی عکس های ما را به 1 تبدیل میکند. هم برای عکس های train و هم برای عکس های validation این تغییر اعمال شده. با غیر رنگی کردم عکس ها در برخی حالت ها عکس های ما بسیار تیره و غیر قابل مشاهده می شوند این به این دلیل است که نور تصاویر کم است.

- برای آنکه نور تصاویر را زیاد تر کنیم از Histogram Equalization استفاده کرده ایم. این تغییر باعث میشود که تصاویر ما بیشتر قابل دیدن باشند و برای شبکه نیز بیشتر قابل فهم میشوند.
  با توجه به دو عکسی که نمایش داده شده، تاثیر Histogram Equalization برروی عکس کاملا واضح و مشخص است.
  همانطور که معلوم است کانال رنگی عکس های ما به 1 کاهش یافته و این به این معنی است که عکس های ما سیاه و سفید شده اند.
  دقیقا این تغییر را بر روی داده های valid نیز اعمال کردیم.

حال شبکه را پس از اعمال این تغییرات بر روی عکس ها train میکنیم، همان مدل قبلی را اعمال کرده و آن را بر روی تصویر تغییر داده شده اعمال میکنیم

همانطور که مشخص است دقت مدل به 95 درصد رسیده پس یعنی اعمال این تغییرات مانند سیاه سفید کردن و تغییر شرایط نوری باعث بهبود شبکه شده و بر روی داده های تست نیز اگر امتحان کنیم درصد خوبی را به درستی تشخیص میدهد.

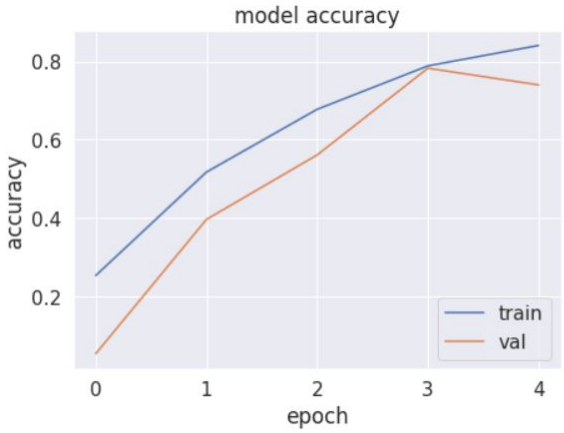


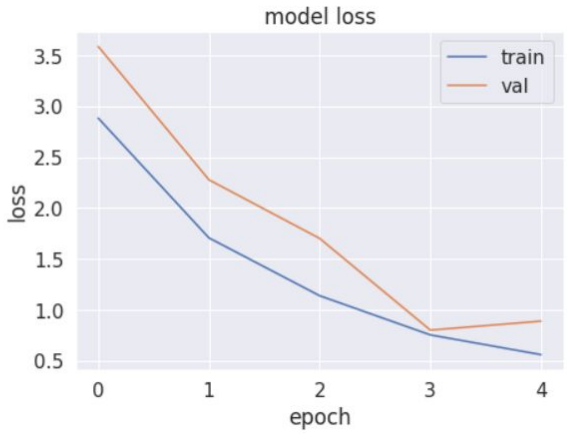


- برای افزایش عکس ها کاری که انجام میدهیم این است که توسط یک data generator برخی تغییرات(مانند zoom کردن و یا چرخش عکس ها و یا شیفت دادن عکس ها به چپ و راست) را بر روی دسته ای از عکس ها اعمال می کنیم، سایز این دسته ها را 5 در نظر گرفتم یعنی در واقع 25 تا عکس جدید به عکس های قدیمی اضافه میشود. سپس این عکس های جدید تولید شده(و همچنین label های جدید) را به X_train و y_train قبلی متصل میکنیم و در واقع آرایه ما زیاد شده و در هر مرتبه با تولید عکس های جدید میتوانیم هر تعداد غکس جدید که بخواهیم به آرایه عکس های قدیمی خود اضافه کنیم.(دوبار این کار رو انجام دادم و به طور رندم 50 تا عکس به عکس های قبلی اضافه کردم)

حال دوباره با زیاد کردن تعداد عکس ها مدل را دوباره compile کرده و نتیجه را مشاهده میکنیم.
زیاد کردن عکس ها را بر روی داده های validation نیز میتوان اعمال کرد اما این کار خیلی نتیجه شبکه را تغییر نداد!

با افزایش تعداد عکس ها وضعیت شبکه بهتر نشد و مقدار دقت آن پایین آمد(به 80% رسید) و همچنین عملکرد بر روی داده های تست نیز کاهش یافت.





- بهترین مدل همان مدلی بود که عکس ها را سیاه سفید و سپس نور آن ها را توسط eq زیاد کرده بودیم، این مدل را بر روی داده های تست نیز پیاده سازی کردم و نتیجه بسیار خوبی داشت. دقت آن بر روی داده های تست به 92 درصد رسید و این یعنی اینکه اکثر تابلو ها را به درستی پیش بینی میکند.
وزن های بهترین مدل و همچنین خود مدل در درایو ذخیره شده و ارسال میگردد.

برای آنکه بتوان فهمید که چه تابلویی با دیگری بیشتر اشتباه گرفته میشود از confusion matrix استفاده کردم همانطور که مشخص است این ماتریس متریک های مختلفی را از جمله precision را به ما نشان میدهد. همانطور که مشخص است اکثر تابلوها نتیجه خوبی (اغلب بالای 70% درست) دارند اما برخی تابلو ها هستند که با دیگر اشتباه گرفته میشوند.

40 تا از تابلو های Roundabout Mandatory به نام No passing پیشبینی شده اند.

48 تا از تابلو های Dangerous curve to right به نام yield پیشبینی شده اند.

27 تا از تابلو های speed limit 20 به نام End of speed 80 پیشبینی شده اند.

تابلوی pedestrians بیشتر از بقیه تابلو ها اشتباه پیش بینی شده است و فقط 31 یی از آن ها درست و به نام اصلی خود پیشبینی شده است. این اشتباه می تواند ناشی از مشخص نبودن تصویر داخل تابلو باشد.

- در قسمت آخر کد نیز خروجی لایه های مختلف شبکه را بر روی عکس دلخواه نمایش داده ایم. این عکس هایی که برای هر لایه نمایش داده شده است در لایه های عمیق تر، کوچک تر میشود و هر کدام ویژگی قسمت های مختلف عکس را خارج میکند که با توجه به آنها شبکه میتواند متوجه عکس ها بشود.
  همچنین این عکس ها نشان دهنده این هستند که هر فیلتر در آن لایه کانولوشنی چگونه عکس ورودی را پروسس کرده است.