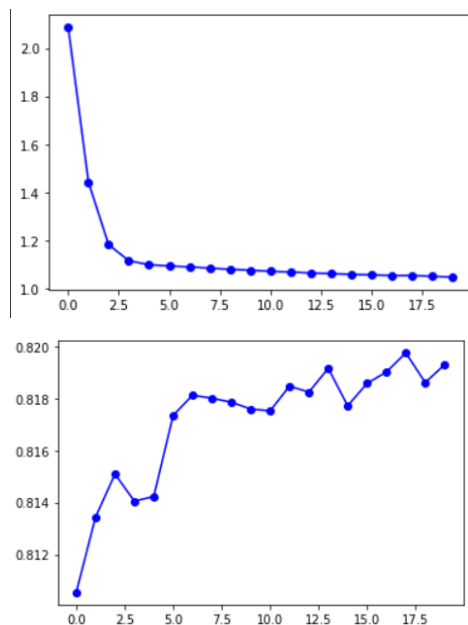


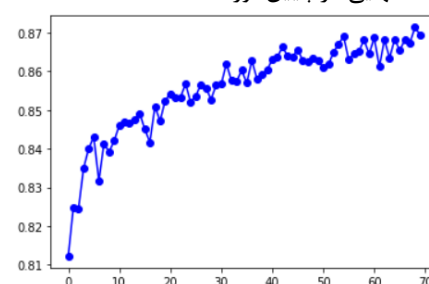
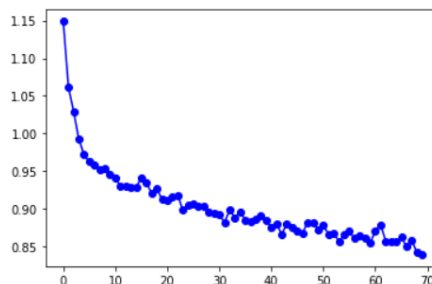
همانطور که میدانیم عکس های بافت شناسی ای که به کمک رنگ های هماتوکسیلین و اتوزین نمونه برداری می شوند یکی از مهم ترین تصاویری هستند که در عرصه تشخیص پاتولوژی های سلولی مورد استفاده قرار میگیرند. شیوه پخش این رنگ های شیمیایی به این گونه میباشد که یکی از آن ها به غشای سلولی که بیشتر حاوی مواد چربی است متصل شده و دیگری به ماده ژنتیکی که در هسته سلول تجمع پیدا کرده متصل می شود. کمپلکس هایی که این ترکیبات شیمیایی ایجاد می کنند باعث می شود این دو ناحیه در زیر میکروسکوپ به دو رنگ متفاوت دیده شوند. از روی این تصاویر که کمک نمونه برداری از بافت به خصوصی از بیمار در آزمایشگاه به وجود می آید، به کمک شکل هسته و سلول و موقعیت و تعداد آن ها پزشک می تواند نوع سلول و احتمال وجود سرطان یا دیگر بیماری های مربوط به بافت به خصوصی را تشخیص دهد. دیتاستی که در این پروژه استفاده شده است لیبیل های متفاوتی را شامل می شود که یکی از آن ها که به نوعی جامع تر می باشد شامل تشخیص مکان دقیق هسته ها و نوع سلول با توجه به شکل هسته می باشد که برای این تمرین از این لیبیل استفاده شده است.

دیتا آگمنتیشن به کمک کراس و به روشی که تا کنون انجام میدادیم یعنی به کمک کلاس `ImageDataGenerator` در اینجا شاید کارا نباشد به این دلیل که در تسک سگمنتیشن ما نیاز داریم که لیبیل هم که در واقع در این تسک خود یک تصویر است همپا با دیتاهای ورودی مدل تغییر کنند که این کار با این کلاس شاید شدنی نباشد یا سخت باشد. برای همین معمولاً از کتابخانه دیگری به نام `imgaug` باید استفاده کرد با توجه به سرچی که انجام دادم ولی ترجیح دادم یک بار بدون آگمنتیشن این تسک را انجام بدهم یک بار و اگر ایرادی در ترین شدن مدل بود یا به مسئله اورفیت برخورد کردم از آگمنتیشن استفاده کنم. به طور کلی با توجه به حساس بودن این تسک و جزئیات ریزی که ممکن است در تصاویر وجود داشته باشد باید دقت کنیم که در ابتدا شکل هسته و مورفولوژی سلول در این تسک از اهمیت زیادی برخوردار می باشد، برای همین بهتر از شیر کردن یا کشیدن تصویر به طور کلی پرهیز کنیم. به غیر از این مورد باید توجه کنیم که رنگ ها در این تسک مهم هستند پس در حالتی که تصویر را به سمت های مختلف حرکت یا شیفت میدهیم باید دقت به `fill_mode` دقت کنیم که رنگی از صفحه به حواشی عکس منتقل نشود.

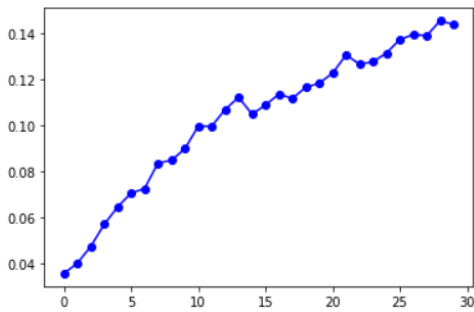
برای این تمرین به علت اشتباهی که داشتیم در انجام کد زدن ترنسفر لرنینگ و اینکه حس میکردم مشکل از مدل هایی است که دانلود کردم یا کتابخانه هایی است که از آن ها استفاده کردم، یک مقدار تعداد مدل ها و کتابخانه هایی که در برنامه ام وجود دارد زیاد هستند که البته به کمک تایتل ها خود کد ها هاید شدند که از بی نظمی جلوگیری شود ولی مدل اصلی همان اولین مدل است که در سایت کگل برای شناسایی سنگ های روی کره ماه ازش استفاده کردند و مدل را بر روی برنامه شان قرار دادند و من هم به کمک ترنسفر لرنینگ از همان مدل استفاده کردم به این علت که حاشیه و شکل سگمنتیشن آن تسک مشابه تشخیص هسته سلول ها بود.



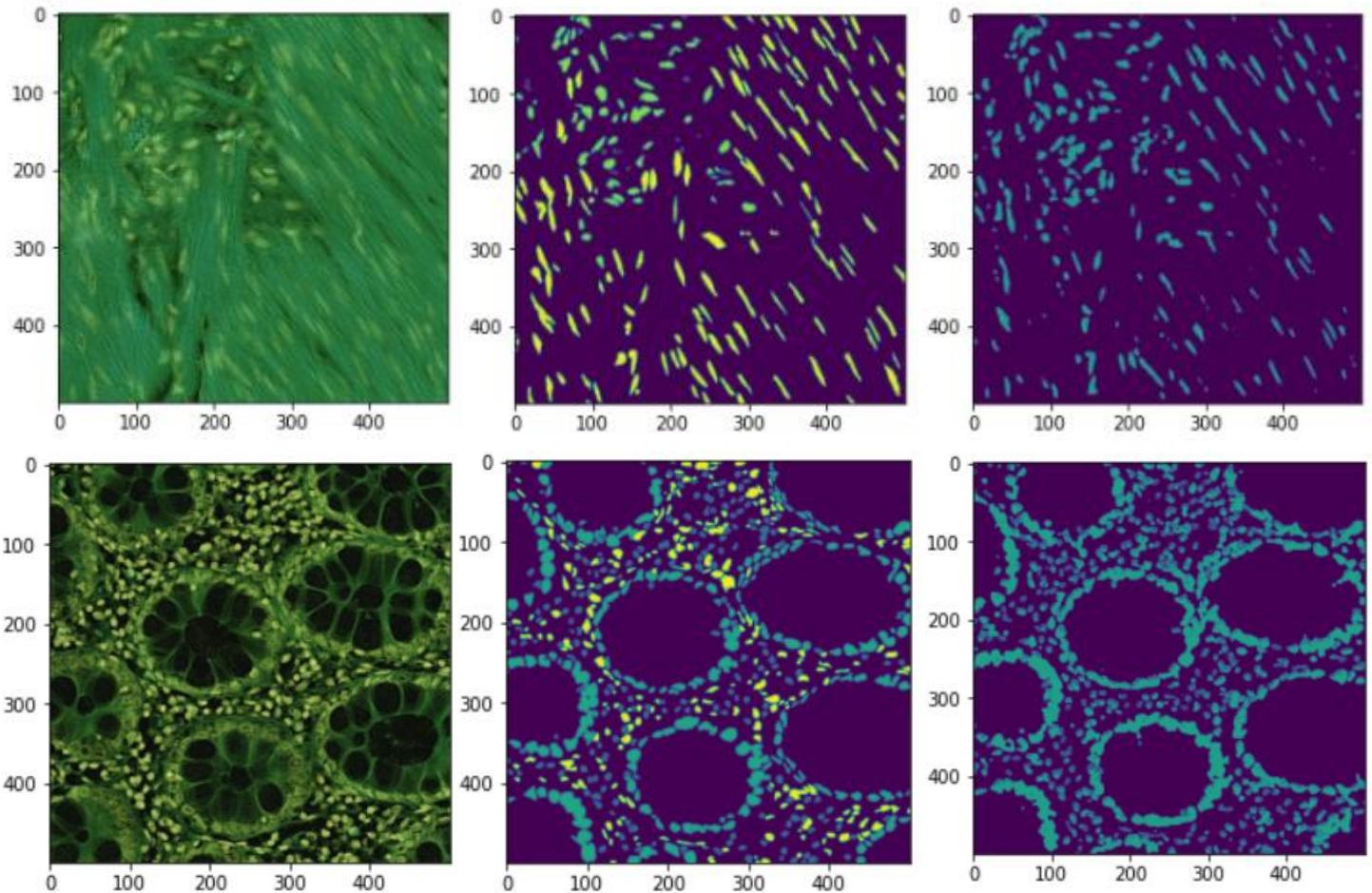
در ابتدا یک لایه کانولوشنی دیگر به انتهای مدل اضافه کردم و ابتدا لایه آخر را ترین کردم که نمودار های لاس و اکیورسی آن را در مقابل میبینیم و سپس کلا مدل را فاین تیون کرده و دقت نهایی را بر روی دیتاست تست امتحان کردیم. لازم به ذکر است که متریک در اینجا پیکسل اکیورسی می باشد و تمامی نمودار ها و دقت نهایی در پایین آورده شده است.



```
1/1 [=====] - 47s 47s/step - loss: 0.7585 - acc: 0.8119
[0.7584843039512634, 0.8119394183158875]
```



حال شاید تصور کنیم که اکپورسی خوب در اینجا به این معنی است که مدل هم عملکرد خوبی را دارد ولی برای تایید این قضیه باید به سراغ معیارهای دیگری برای ارزیابی مدلمان باشیم. به همین علت بار دیگر مدل را ترین کردم و این بار از دو معیار  $IoU$  و  $MeanIoU$  استفاده کرد.  $MeanIoU$  تقریباً تغییری نکرد در طول آموزش ولی همانطور که میبینم  $IoU$  یک روند افزایشی داشته است ولی همچنان مقدار آن خیلی قابل توجه نمی‌باشد. با این وجود برای اینکه ببینیم عملکرد مدل به چه شکل است یک بار عکس اصلی به همراه لیبل و به همراه لیبل پردیکت شده را در یک جا می‌آوریم که در زیر قابل ملاحظه می‌باشد. (دقت شود سری تصویر اول در پایین یکی از تصاویر دیتاست تست می‌باشد و ردیف دوم مربوط به ترین می‌باشد).

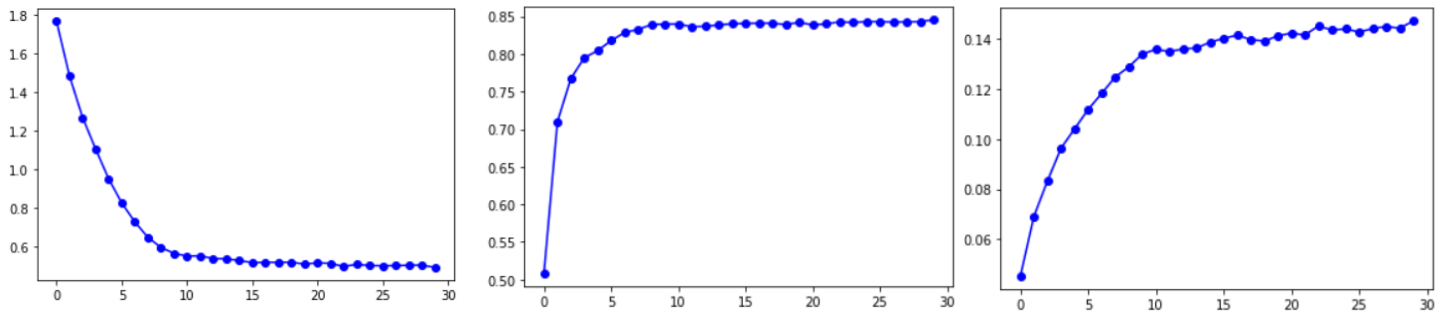


	0	1	2	3	4	5	6	7
0	147866.0	411.0	2758.0	1222.0	38629.0	585.0	1.0	5.0
1	648.0	82.0	599.0	299.0	3404.0	63.0	1.0	0.0
2	504.0	86.0	808.0	346.0	4074.0	81.0	0.0	0.0
3	333.0	69.0	1613.0	518.0	8965.0	102.0	0.0	0.0
4	628.0	226.0	3552.0	922.0	28475.0	282.0	0.0	20.0
5	59.0	91.0	372.0	82.0	1207.0	12.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

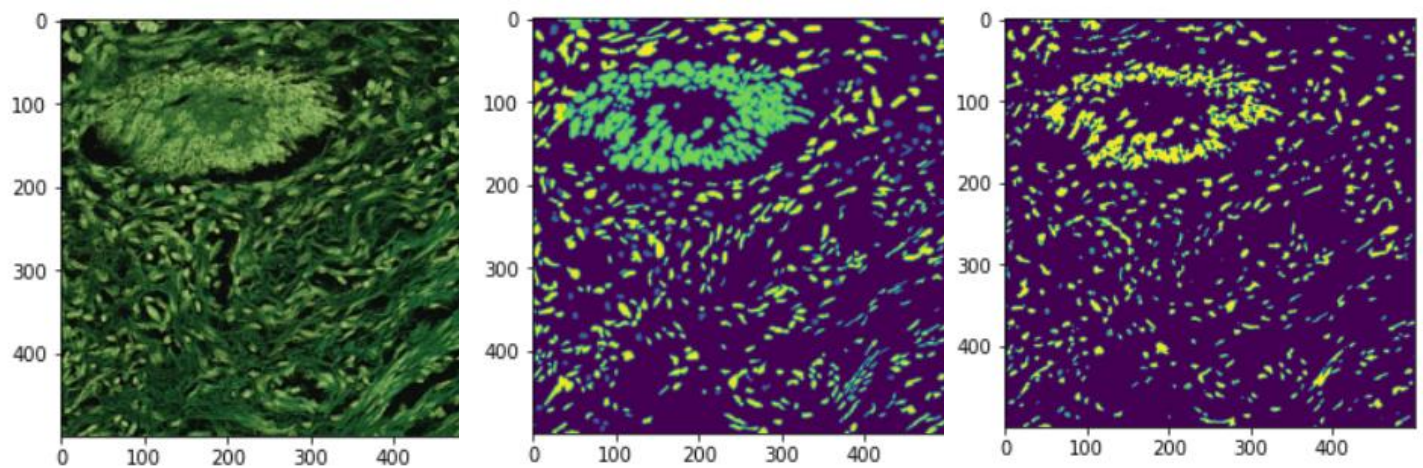
خب همانطور که ملاحظه می‌کنیم عملکرد مدل شاید مقداری در تشخیص جایگاه خوب عمل کرده باشد ولی در تشخیص کلاس حداقل میتوان گفت که عملکرد چندان مناسبی ندارد. برای اینکه دید بهتری داشته باشیم نسبت به عملکرد مدل بر روی کلاس‌های مختلف از کانفیوژن ماتریکس بر روی دیتاست تست استفاده می‌کنیم که در رو به رو مشاهده میکنیم جدول مربوط به آن را. همانطور که مشهود است یک سری از کلاس‌ها مثل کلاس ۶ و ۷ اصلاً عملکرد خوبی ندارند که به نظر می‌رسد به خاطر کمبود دیتا می‌باشد. به همین دلیل به سراغ دیتا آگمنتیشن می‌رویم که همانطور که گفته شد به دلیل اینکه این پروژه یک پروژه‌ی سگمنتیشن می‌باشد از خود کتابخانه کراس به سختی میتوان استفاده کرد برای این پیش پردازش و به سراغ یکی دیگر

از کتابخانه‌هایی که مخصوص این کار میباشد می‌رویم و ملاحظاتی که گفته شد در ابتدای گزارش را رعایت می‌کنیم. در این جا اطلاعات مربوط به مدل پس از استفاده از آگمنتیشن را میبینیم.





تصاویری که از مدل گرفته شد این بار میتوان گفت در کلاس های مختلف نیز بد عمل نکرد و در تصاویر خروجی تا حدودی البته با دقت نه چندان زیاد کلاس ها نیز تشخیص داده شده‌اند.

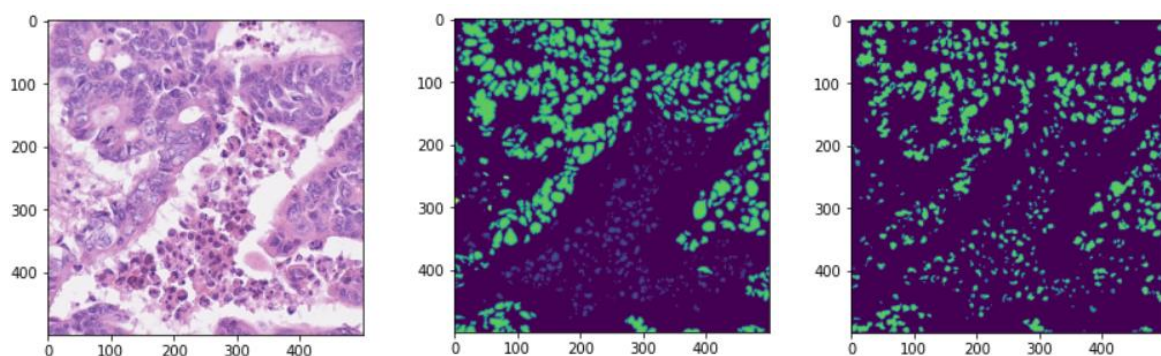


	0	1	2	3	4	5	6	7
0	181000.0	2.0	399.0	0.0	10076.0	0.0	0.0	0.0
1	3328.0	0.0	130.0	0.0	1638.0	0.0	0.0	0.0
2	3462.0	0.0	121.0	0.0	2316.0	0.0	0.0	0.0
3	4059.0	0.0	262.0	0.0	7279.0	0.0	0.0	0.0
4	9922.0	0.0	997.0	0.0	23186.0	0.0	0.0	0.0
5	277.0	0.0	204.0	0.0	1342.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

با توجه به کانفیوژن ماتریکس که در رو به رو آمده است این بار وضعیت مدل همچنان بهتر از حالات قبلی نشده است و همچنان حداقل در تشخیص کلاس های کم جمعیت مشکل اساسی دارید.

برای آنبالانس بودن این دیتاست اندکی در توضیحات سایت خود دیتاست آمده است که برای اینکه مقداری عملکرد مدل بهبود پیدا کند کلاس های 3 و 4 یکی کرده و در کلاس سلول های اپیتلیالی و کلاس های 5 و 6 و 7 را در کلاس سلول های spindle shape قرار داده است. به امید اینکه مقداری عملکرد مدل بهتر شود از این روش نیز یکبار استفاده میکنیم.

خب با توجه به دقتی که بر روی دیتاست تست و تصاویر حاصل از مدل میتوان نتیجه گرفت این بار دقت و عملکرد مدل به مراتب بهتر از حالات قبلی شده است و عملکرد قابل قبولی بر روی دیتاست ولیدیشن ما دارد.



1/1 [=====] - 49s 49s/step - loss: 0.4660 - acc: 0.8420 - iou\_score: 0.2579 - mean\_io\_u: 0.4000