

بسم الله الرحمن الرحيم

دید کامپیوتری

خانم دکتر هدی محمد زاده

گزارش بخش کامپیوتری تمرین سری 3

امیرحسین رستمی ۹۶۱۰۱۶۳۵

دانشگاه صنعتی شریف

بهار ۹۹

گزارش بخش کامپیوتری (گزارش بخش های تعوری و پژوهشی در فایل reportTheory.pdf آورده شده است).

فایل Jupyter Notebook در فایل پاسخ قرار داده شده است و در این قسمت کد های هر بخش به همراه کامنت هایی به مقدار کافی و خروجی ها هر بخش آورده شده است و لذا طبق صورت تمرین دیگر به نوشتن مفصلی از گزارش نمی پردازیم و فقط برخی توصیفات کیفی را در اینجا بیان می کنیم.

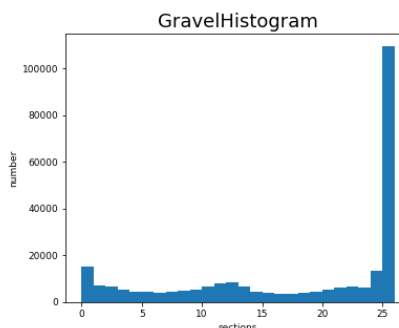
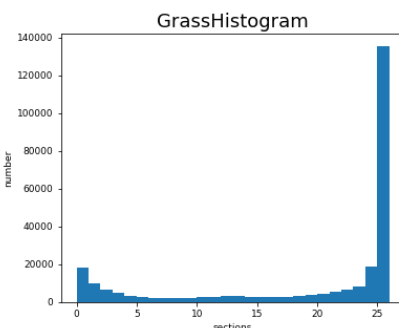
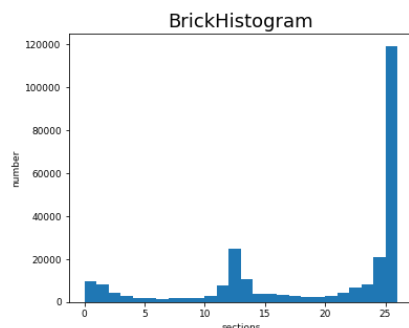
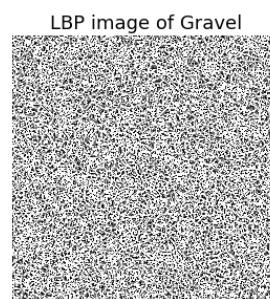
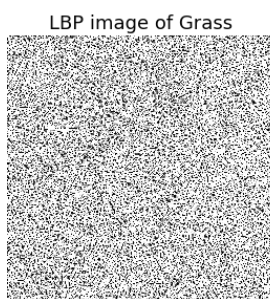
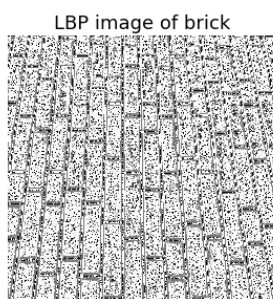
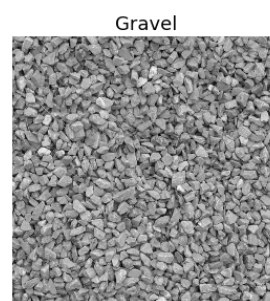
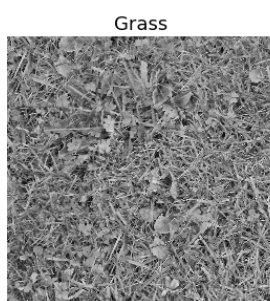
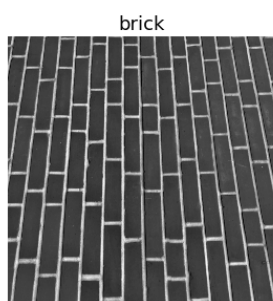
سوال اول:

با بررسی و اعمال تغییرات مکرر روی پارامتر های توابعی اعم از detectMultiScale و ... به مقادیری برای متغیر ها رسیدیم که برای هر تصویر خروجی های کاملاً مقبولی را فراهم می کند جهت مشاهده ی خروجی هر تصویر و پارامتر های تعیین شده لطفاً به فایل Jupyter Note مراجعه کنید.

سوال دوم:

با بررسی آرگومان های موجود و تغییرات آن ها سرانجام با تعیین پارامتر های یکسانی برای هر سه عکس به نتایج زیر رسیدیم:

جهت مشاهده پارامترها به فایل Jupyter Note مراجعه کنید.



همانطور که ملاحظه می کنید سه عکس در نمودار های هیستوگرام در sections بازه ی 10 تا 15 با یکدیگر متفاوت اند و این به دلیل تفاوت آن ها در درشتی المان های موجود در عکس است و با تعیین ترشهولد ی مناسب می توان آن ها را از یکدیگر تمیز داد.

درشتی: آجر < ماسه < چمن

اندازه تراکم در Section های بین 10 تا 15 : $Hist(grass) < Hist(gravel) < Hist(brick)$

سوال سوم: (در این سوال خروجی ها و کامنت های زیادی در فایل Jupiter Note قرار داده شده است، لطفا به آن حتما مراجعه کنید).

قسمت الف:

خروجی و توضیحات در فایل Jupiter Note.

قسمت ب:

Dropout:

برای جلوگیری از رخ داد overfitting وقتا دیتا ست و تعداد پارامتر ها فاصله ی زیادی از هم داشته باشند نیاز است تا در لایه ها با ضربی به نام dropout (همانطور که از اسمش پیداست) وابستگی در برخی قسمت ها به لایه های قبل را از بین برد (اثرشان را صفر میکنیم در تعیین لایه ی بعدی) و این چنین شبکه دیگر خیلی به همه ی داده های train حساس نمی گردد و رخداد overfitting کاهش پیدا میکند. اما چند نکته ...

1- ضرب dropout نباید خیلی کم باشد چون در این صورت اثرش در کاهش overfitting کم خواهد بود.

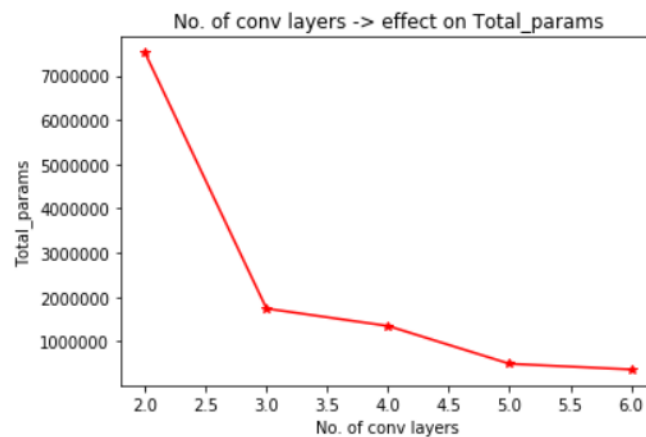
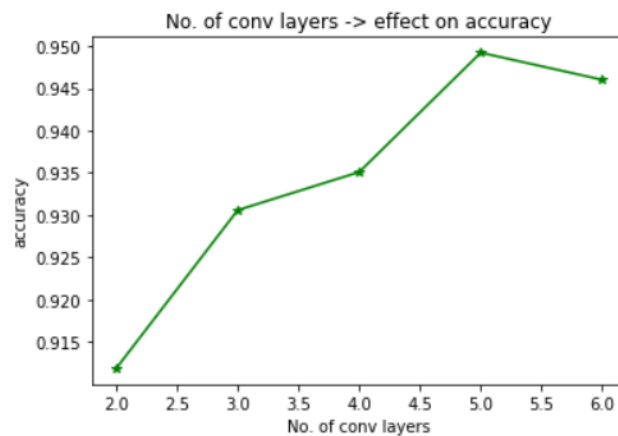
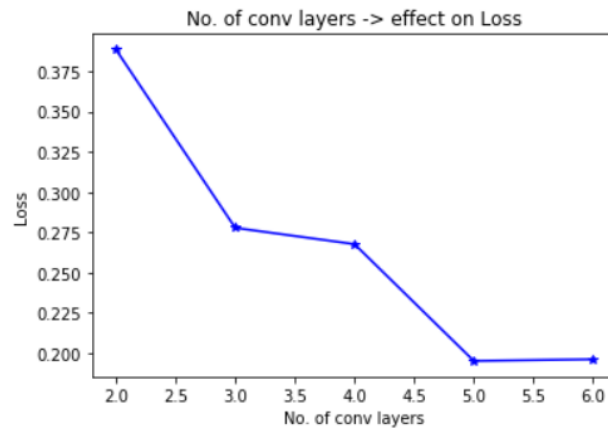
2- ضرب dropout نباید آنقدر هم زیاد باشد که دقت شبکه را کاهش دهد.

لذا باید با در نظر گرفتن هر دو شرط 1 و 2 به ضربی بهینه در هر دو مورد برسیم که با بررسی بنده ضرایب 0.25 و 0.5 در برخی لایه ها خوب بودند.

بهترین شبکه:

دقت کنید که فیلتر های کانولوشنی باید رزونانسی از هر دو کرنل های 3×3 و 5×5 باشد و استفاده فقط! از هر کدام خوب نیست چون کرنل های سائز کوچک اطلاعات کمی حمل می کنند و برای برخی کاربرد ها خوب اند و کرنل های بزرگ اطلاعات بیشتری حمل می کنند و برای برخی کاربرد ها خوب اند و نمی توان گفت کرنل بزرگتر لزوما برتر از کرنل کوچکتر است چرا که استفاده ی زیاد از کرنل های بزرگ متوالی باعث رخ داد overfitting می شود و از طرفی استفاده انحصاری از کرنل های کوچک هم دقت کمتری خواهد داشت و طول لایه ها را برای رسیدن به مقصدی خاص زیاد می کند لذا باید با رزونانسی از هر دو کرنل در شبکه را train کرد.

مقایسه ی پارامتر های Loss, Accuracy, ParamsNum :



نکته: همانطور که ملاحظه می کنید در حرکت از تعداد لایه های کانولوشنی از 5 به 6 مشاهده ی افزایش خطا و کاهش accuracy می شویم که این به دلیل رخ داد overfitting است، بله درست است که در لایه های کانولوشنی رخ داد overfitting نسبت به حالت fullyConnected کاهش پیدا می کند اما بالاخره افزایش لایه ها از یه حد بیشتر به هرحال باعث رخ داد این واقعه می شود.

الباقی مقایسه ها به زیبایی در نمودار ها مشخص است و از شرح توضیح اضافی پرهیز میکنیم.

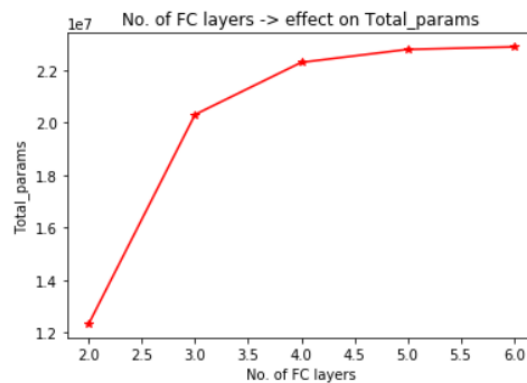
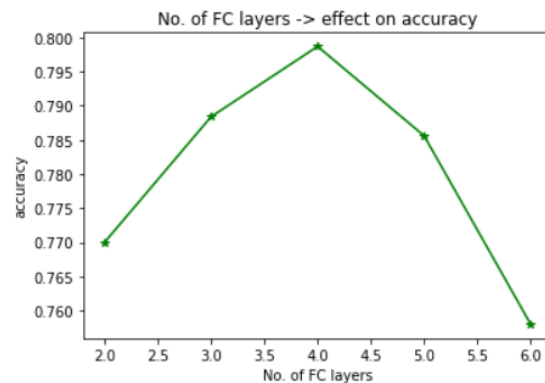
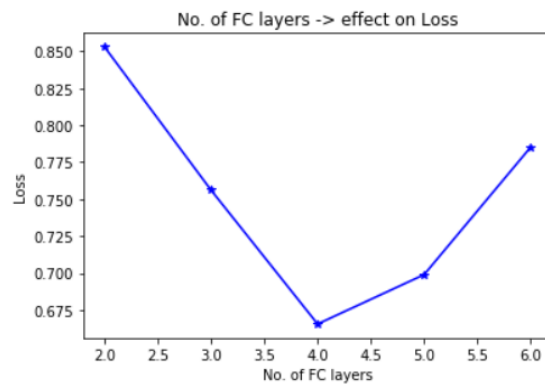
قسمت پ:

پیش از بررسی نمودار ها انتظار می رود که

1- در شبکه ها fullyConnected رخ داد overfitting به ازای افزایش لایه های بینابینی نسبت به حالت کانولوشنی زودتر رخ بدهد.

2- میزان دقت در قیاس با شبکه ی کانولوشنی مشابه کمتر باشد و خطایش بیشتر باشد.

حال به مشاهده نمودار ها می پردازیم:



نتایج:

1- همانطور که مشاهده می کنید به ازای افزایش لایه از 4 به 5 overfitting رخ داد (با مشتق زیاد) و این در حالی است که در حالت شبکه ی کانولوشنی به ازای حرکت از تعداد لایه های 5 به 6 آن هم با مشتقی overfitting بسیار اندکی نسبت به این حالت پدیده overfitting رخ داده است (تایید پیش بینی 1)

2- همانطور که مشاهده می کنید میزان Loss و accuracy در دو حالت FullyConnected و کانولوشنی اختلاف فاحشی با یکدیگر دارند و به وضوح کیفیت و دقت حالت کانولوشنی نسبت به حالت FC بیشتر است (تایید پیش بینی 2) و خب این منطقی است چون در FC تعداد زیادی پارامتر های اضافی به خاطر fullyConnected بودن استفاده شده است که سبب حساسیت زیاد به داده های train و کاهش دقت در تشخیص داده های test شده است.

قسمت ت:

در پاسخ Jupyter Note ضمیمه شده آورده شده است.

قسمت ث:

در پاسخ ضمیمه شده 5 عکس یکسان را در کانولوشنی و fullyConnected تست کرده ایم (خروجی ها در فایل Jupyter Note موجود است) و به تفاوت فاحش دقت تشخیص (تایید مجدد نتایج قسمت پ) رسیدیم و دیدیم که لایه های کانولوشنی با دقتی بسیار بیشتر عکس هارا تشخیص دادند.

جهت مشاهده ی گزارش بخش های تعوری و پژوهشی به فایل *reportTheory* مراجعه کنید.