قسمت اول) QATM:

روش QATM یکی از روشهای تطبیق کلیشه است که در ادامه به توضیح آن پرداخته شده­است. این روش بر اساس میزان تمایز(فاصله) کلیشه از تصویر جستجو پیاده­سازی شده­است که سناریوهای تطبیق مختلف را نیز در محاسبه در نظر می­گیرد مانند 1 به 1 یا چند به یک و...

یکی از مزیت­های مهم این روش این است که به طوری طراحی و پیاده­سازی شده­است که می­توان آنرا به صورت یک لایه قابل آموزش در شبکه­های عمیق بکار برد(تمام عملیات­ها مشتق­پذیر است).

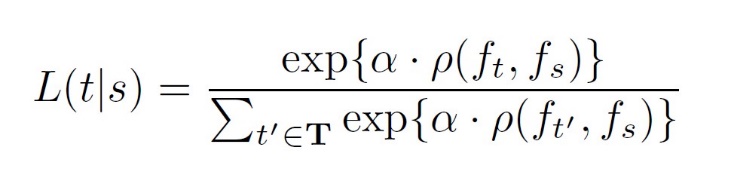
در محاسبات این روش تمامی پیکسل­های کلیشه و قسمت کاندید تاثیرگذار هستند.

محاسبات این روش به ترتیب عبارت است از:

در مرحله اول با استفاده از یک شبکه از پیش آموزش دیده از عکس جستجو و کلیشه ویژگی استخراج می­شود نکته حائز اهمیت این است که مشابه شبکه­های Siamese این دو شبکه با یکدیگر یکسان هستند.

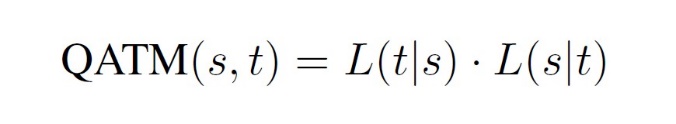
در مرحله دوم نیاز به محاسبه معیار فاصله بر روی ویژگی­های بدست آمده داریم که در این قسمت از cosine similarity استفاده شده­است. این فاصله به صورت patch-wise محاسبه می­شود یعنی فواصل کلیشه با تمام ناحیه­های کاندید تصویر جستجو محاسبه می­شود.

در مرحله سوم تابع softmax بر روی فواصل محاسبه شده اعمال می­گردد تا مقادیر فرم احتمالاتی به خورد بگیرند. در این حالت مطلوب مسئله این است که در حالت تطبیق مقادیر به 1 نزدیک شوند و در حالت عدم تطبیق این مقادیر به صفر میل کنند. به همین دلیل در محاسبه تابع softmax یک ضریب آلفا دخیل شده تا این مورد را تنظیم کند. مقدار آلفا هم می­تواند در فرآیند آموزش به عنوان یک پارامتر یاد گرفته شود و یا به صورت یک مقدار ثابت اعمال شود. فرمول محاسبه این قسمت به شرح زیر است.

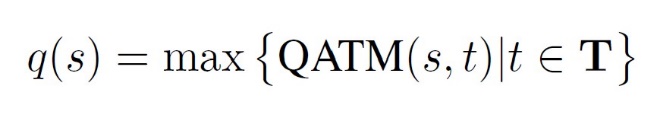


L(t|s) نشان دهنده مقدار احتمال تطبیق تمام کلیشه­ها بر روی یک تصویر جستجو است به همین ترتیب نیز L(s|t) محاسبه می­شود که نشان دهنده مقدار احتمال تطبیق یک کلیشه بر روی تمام تصاویر جستجو است.

با ضرب دو مقدار احتمال محاسبه شده بالا، مقدار QATM به صورت زیر محاسبه می­شود.



در مرحله چهارم بررسی می­شود که به ازای هر تصویر جستجو، چه کلیشه­ای بهترین مقدار را به خود اختصاص داده است تا آنرا به عنوان کلیشه انتخاب شده برای عکس در نظر بگیرد.



در مرحله پنجم بررسی می­شود که کلیشه انتخاب شده برای عکس در کدام ناحیه از تصویر جستجو بیشترین مقدار تطبیق را به خود اختصاص داده است و آن ناحیه به عنوان ناحیه برتر انتخاب می­شود.

پیاده­سازی:

ابتدا تابع creat\_model تعریف می­شود که شبکه­ی از پیش آموزش دیده شده­ای را که برای استخراج ویژگی استفاده می­شود به همراه مقدار آلفا به عنوان پارامتر ورودی دریافت می­کند. در ابتدا لایه­های ورودی برای هریک از تصویر جستجو و کلیشه تعریف می­شود و سپس با استفاده از شبکه مذکور از آن دو ویژگی استخراج می­شود. در مرحله بعدی با استفاده از لایه MyNormLayer مقادیر ویژگی نرمال می­شود. سپس مقادیر نرمال شده به عنوان ورودی به لایه شخصی­سازی شده Lambda داده می­شود که در آن با استفاده از متد einsum مقادیر cosine distance به صورت patch-wise محاسبه می­شود. در نهایت خروجی این قسمت به لایه QATM داده می­شود که در نهایت Hitmap مورد نظر بر روی تصویر جستجو بدست می­آید.

در قسمت بعدی تابع model\_eval تعریف می­شود که دو شبکه استخراج ویژگی(یک شبکه برای تصاویر با سایز بزرگ و دیگری برای تصاویر با سایز کوچک) را به همراه آلفا به عنوان ورودی دریافت می­کند. بر اساس سایز تصویر( بزرگ یا کوچک) شبکه استخراج ویژگی متناظر به تابع creat\_model داده می­شود تا مدل متناظر با آن ساخته شود. با استفاده از فراخوانی متد predict بر روی تمامی تصاویر و کلیشه­های متناظر با آن، hitmap مورد نظر محاسبه و ذخیره می­شود.

در مرحله بعدی با استفاده از متد compute\_score امتیاز قسمت­های مختلف تصویر را بدست می­آوریم و در نهایت ground truth مربوط به تصویر به همراه امتیاز بدست آمده return می­شود.

سپس متد all\_sample\_iou بر روی امتیازات بدست آمده اعمال می­شود که در آن با استفاده از تابع کمکی locate\_bbox ابتدا بر روی نقشه امتیازات max را پیدا کرده و در مرکز box به اندازه template قرار می­دهد. در نهایت IoU باکس بدست­آمده با ground truth محاسبه می­شود و مقادیر IoU محاسبه شده خروجی داده می­شوند.

در مرحله آخر با استفاده از تابع score2curve مقادیر TP و FP برای مقادیر آستانه مختلف محاسبه می­شود تا در نهایت ROC curve با استفاده از این مقادیر رسم شود. با استفاده از این نمودار AUC نیز محاسبه شده و گزارش می­شود.

قسمت دوم) مقاله انتخابی

در این مقاله روش پیشنهادی در ابتدا سعی در پیدا کردن قسمت کلیدی در کلیشه دارد (key part). به صورت پیشفرض ابعاد قسمت کلیدی را به اندازه نصف کلیشه اصلی در نظر می­گیرد. به صورت پنجره لغزان با stride ای به اندازه 4/1 اندازه کلیشه قسمت کاندید را بر روی کلیشه در نظر می­گیرد و آنرا به مقدار میانگین پیکسل­های کلیشه جایگزین می­کند. حال از بین کاندیدهای تولید شده، کاندیدی را انتخاب می­کند که مقدار ضرب داخلی آن با تصویر اصلی کلیشه مینیمم باشد. این محاسبه به این معنی است که با ماسک کردن آن قسمت اطلاعات بیشتری از دست رفته است پس آن قسمت از اهمیت بالاتری برخوردار بوده و به عنوان قسمت کلیدی انتخاب می­شود.

حال در نظر داریم تا تاثیر این بخش از کلیشه را بر بقیه قسمت­های آن برتری ببخشیم به صورتی که به ویژگی­های استخراج شده مربوط به قسمت کلیدی نسبت به قسمت­های دیگر وزن بیشتری اختصاص دهیم. برای اینکه به این قسمت وزن بیشتری اختصاص دهیم، ابتدا ویژگی­های قسمت کلیدی و کلیشه را با استفاده از شبکه pretrained استخراج می­کنیم. سپس cross correlation میان آن دو را محاسبه می­کنیم(قسمت کلیدی را به عنوان فیلتر در نظر می­گیریم). نتیجه این قسمت را با ویژگی­های استخراج شده از کلیشه ترکیب می­کنیم.

پیاده­سازی:

برای پیاده­سازی از همان شبکه از پیش آموزش دیده شده­ای که در QATM استفاده شده­بود، بهره برده­ایم. ایده این مقاله به صورت مجزا قبل از اعمال الگوریتم QATM بر روی کلیشه اعمال شده­است بطوریکه کلیشه وزن دار شده به عنوان ورودی الگوریتم QATM داده می­شود.

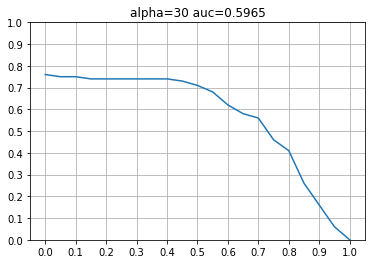
در قسمت اول تابع find\_keypart پیاده­سازی شده­است بطوریکه کلیشه را به عنوان ورودی دریافت می­کند. سپس تمامی کاندید­های مورد نظر را که به تعداد 9 عدد هستند تولید می­کند. تک تک کاندیدها با کلیشه اصلی ضرب داخلی می­شوند که به نوعی یک معیار محاسبه شباهت می­باشد. حال کاندیدی که کمترین میزان شباهت را دارد به عنوان کاندید برتر انتخاب می­شود. سپس از قسمت کلیدی، کلیشه و تصویر جستجو با استفاده از شبکه مورد نظر(بخشی از vgg19) ویژگی استخراج می­شود.

در قسمت بعدی تابع the\_corr پیاده­سازی شده­است که وظیفه محاسبه correlation را بین ویژگی­های قسمت کلیدی و کلیشه را به صورت channel-wise بر عهده دارد که در این محاسبه، ویژگی­های قسمت کلیدی به عنوان فیلتر در نظر گرفته شده­است.

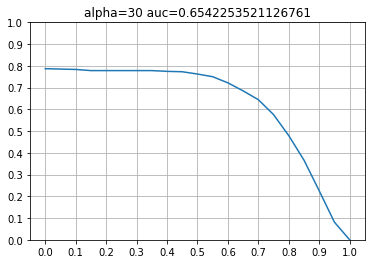
حال برای وزن­دار کردن قسمت کلیدی در کلیشه و همینطور کاهش اهمیت قسمت­های دیگر نیاز داریم تا با روشی این موضوع را اعمال کنیم. برای انجام این قسمت در ابتدا مقادیر correlation را با استفاده از ماژول MinMaxScaler استاندارد می­کنیم(مقادیر بین صفر و یک) و مقدار بدست آمده را 0.75 افزایش می­دهیم که علت آن این است که در بهترین حالت وزن­های قسمت کلیدی 75% افزایش یابد(ضرب در 1.75 شود) و در بدترین حالت وزن­های دورترین قسمت­ها از ناحیه کلیدی 25% کاهش یابد(ضرب در 0.75 شود). اعمال ضرب مذکور با استفاده از یک لایه Multiply() کتابخانه keras پیاده­سازی شده­است.

قسمت سوم) نتایج:

مقایسه نتایج QATM و روش پیشنهادی(100 داده)

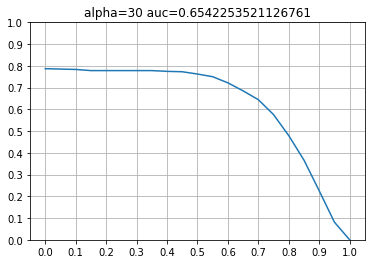


نتیجه QATM

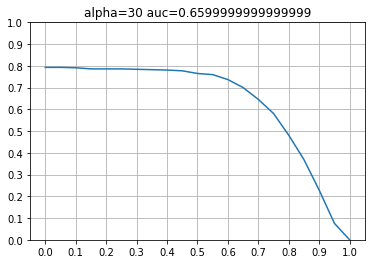


نتیجه روش پیشنهادی

مقایسه نتایج QATM و روش پیشنهادی(500 داده)

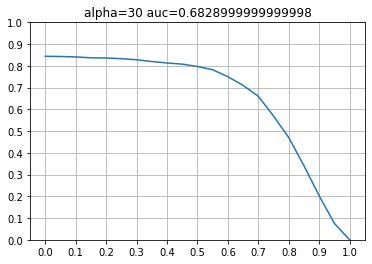


نتیجه QATM

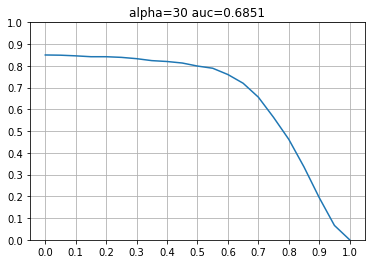


نتیجه روش پیشنهادی

مقایسه نتایج QATM و روش پیشنهادی(1000 داده)



نتیجه QATM



نتیجه روش پیشنهادی