بسم الله الرحمن الرحیم

دانشگاه علم و صنعت ایران

زمستان ۱۳۹۹

مبانی بینایی کامپیوتر پاسخ تمرین سری پانزدهم

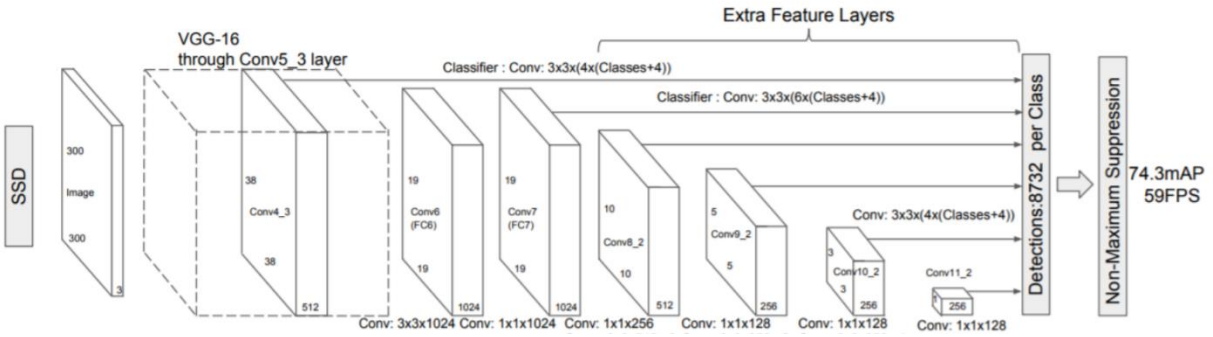
---

۱. شبکه‌های خانواده RCNN را با شبکه‌های YOLO و SSD مقایسه کنید و نقاط ضعف و قوت هرکدام را بیان کنید.

روش‌های خانواده RCNN از دو مرحله تشکیل شده‌اند. مرحله اول، پیدا کردن نواحی کاندیدا است و مرحله دوم، طبقه بندی و رگرسیون نواحی پیدا شده است. در روش RCNN ابتدا نواحی کاندیدا با روش‌هایی مانند selective search استخراج شده و برای استخراج ویژگی، هر ناحیه کاندیدا وارد شبکه CNN می‌شود و بعد از استخراج ویژگی، طبقه بندی ناحیه و رگرسیون box انجام می‌گیرد. در روش Fast RCNN ابتدا برای استخراج ویژگی، تصویر به شبکه CNN داده شده و از feature map خروجی با استفاده از selective search نواحی کاندیدا استخراج می‌گردند (ROI projection با استفاده از تصویر اصلی و نسبت تصویر اصلی به feature map این جستجو انجام و نواحی کاندیدا به دست می‌آیند). در مرحله بعد نیز عمل دسته بندی و رگرسیون با استفاده از SVM انجام می‌گیرد. در شبکه faster RCNN بخش مربوط به selective search نیز حذف شد و به جای آن از شبکه RPN برای استخراج ویژگی‌ها استفاده شده که امکان آموزش انتها به انتهای شبکه را به ما می‌دهد. استفاده از selective search زمانبر بوده و ۲۰۰۰ ناحیه تولید می‌کند که بسیار زیاد است. استفاده از این روش‌ها موجب افزایش سرعت و کاهش استفاده از منابع می‌گردد. شبکه‌های YOLO و SSD نیز شباهت زیادی به faster RCNN دارند از این جهت که در آنها نیز از anchor box ها استفاده می‌گردد (بخش RPN در شبکه faster RCNN) و از ایده پنجره لغزان استفاده می‌کنند ولی تفاوت عمده آن‌ها با این شبکه، رگرسیون و طبقه بندی همزمان است که با بررسی نواحی تصویر است. YOLO در تشخیص اشیا کوچک و همچنین aspect ratio های متنوع (به دلیل استفاده از anchor های محدود) ضعیف عمل می‌کند. شبکه SSD از ایده استفاده از ویژگی‌ها در مقیاس‌های متفاوت استفاده می‌کند و با استفاده از aspect ratio و مقیاس‌های متنوع، نواحی بیشتری را نسبت به YOLO بررسی می‌کند. روش‌های خانواده RCNN از نظر تشخیص اشیا، بسیار قدرتمند بوده ولی حتی سریع‌ترین روش آن یعنی faster RCNN را نیز نمی‌توان به صورت بلادرنگ استفاده کرد. روش‌های single shot مانند YOLO و SSD از نظر زمانی بسیار مناسب هستند ولی از نظر دقت پایین‌تر از faster RCNN قرار می‌گیرند هرچند SSD در دقت پیشرفت قابل توجهی داشته است.

۲.

الف) در تصویر زیر نشان داده شده است که برای هر کلاس، ۸۷۳۲ ناحیه از تصویر ورودی بررسی می‌شود. نحوه محاسبه این عدد را به طور دقیق یادداشت کنید.



در هر feature map به تعداد تمامی پیکسل‌ها و به تعداد anchor box ها، ناحیه مورد بررسی داریم. در مقیاس‌های مختلف، ابعاد ویژگی‌ها از قرار تصویر داده شده در صورت سوال است. طبق توضیحات داریم:

$$(38 \times 38) \times 4 + (19 \times 19) \times 6 + (10 \times 10) \times 6 + (5 \times 5) \times 6 + (3 \times 3) \times 4 + (1 \times 1) \times 4 = 8732$$

ب) در صورتیکه ابعاد تصویر ورودی این شبکه برابر با ۶۰۰×۳۰۰ باشد، تعداد ناحیه‌های مورد بررسی چه عددی خواهد بود؟

با دادن ابعاد ۶۰۰ × ۳۰۰ به شبکه SSD خروجی مدنظر Conv5_3 برابر با ۷۵ × ۳۸ می‌گردد.

مرحله بعد ابعاد لایه FC7 برابر با ۳۸ × ۱۹ می‌گردد (Conv7).

بعد یک لایه کانوولوشن ۳ × ۳ با stride2 پس از یک کاهش بعد با کانوولوشن ۱ × ۱ خواهیم داشت.

پس خروجی این لایه ۱۹ × ۱۰ است (Conv8_2).

در مرحله بعد هم یک کاهش بعد با کانوولوشن ۱ × ۱ داشته و بعد از آن یک کانوولوشن ۳ × ۳ با Stride2 روی آن اعمال می‌گردد و ابعاد نهایی ۱۰ × ۵ می‌گردد (Conv9_2).

در مرحله بعد هم یک کاهش بعد با کانوولوشن ۱ × ۱ داشته و بعد از آن یک کانولوشن ۳ × ۳ با Stride1 روی آن اعمال می‌گردد و ابعاد نهایی ۸ × ۳ می‌گردد (Conv10_2).

در مرحله بعد هم یک کاهش بعد با کانوولوشن ۱ × ۱ داشته و بعد از آن یک کانوولوشن ۳ × ۳ با Stride1 روی آن اعمال می‌گردد و ابعاد نهایی ۶ × ۱ می‌گردد (Conv11_2).

پس تعداد نواحی مورد بررسی از قرار زیر است:

$$(38 \times 75) \times 4 + (19 \times 38) \times 6 + (10 \times 19) \times 6 + (5 \times 10) \times 6 + (3 \times 8) \times 4 + (1 \times 6) \times 4 = 17292$$

.۳

الف) یکی از شبکه‌های backbone که در بسیاری از شبکه‌های تشخیص شی (ازجمله SSD) استفاده می‌شود، شبکه VGG16 است که جزئیات آن در جدول زیر نشان داده شده است. در صورتیکه لایه‌های کاملا متصل این شبکه را به لایه‌های کانوولوشنی (بدون padding) تبدیل کنیم، و در ورودی شبکه یک تصویر ۵۱۲×۵۱۲ قرار دهیم، خروجی شبکه چه ابعادی خواهد داشت؟

در ورودی لایه کاملا متصل یک feature map با اندازه ۷ در ۷ وجود دارد. برای تبدیل این شبکه به شبکه کاملا کانوولوشنی، لایه اول کاملا متصل به یک لایه کانوولوشنی با اندازه کرنل ۷ × ۷ تغییر می‌کند. سپس در لایه‌های بعدی به جای لایه‌های کاملا متصل از لایه‌های کانوولوشنی با اندازه کرنل ۱ × ۱ استفاده می‌شود که تاثیری در اندازه نقشه ویژگی ندارند.

در این شبکه کانوولوشن‌ها با stride یک و padding هستند پس ابعاد را تغییر نمی‌دهند. تنها جایی که stride تغییر می‌کند در pooling ها است. ۵ لایه pooling موجود است که هر کدام دو واحد stride دارند. پس ابعاد نهایی خروجی نهایی قسمت کانوولوشنی vgg16 از قرار زیر است:

$$\frac{512}{2^5} = 16$$

سپس، این خروجی به لایه ۷ در ۷ بدون padding وارد می‌شود. در نتیجه ابعاد خروجی از قرار زیر است:

$$16 - 7 + 1 = 10$$

پس ابعاد خروجی ۱۰ × ۱۰ است.

ب) لایه ابتدایی این شبکه شامل ۶۴ فیلتر ۳×۳ است. در حالتی که ورودی شبکه ۵۱۲×۵۱۲ باشد، این لایه چه تعداد ضرب و جمع نیاز دارد (به طور دقیق محاسبه کنید و مراحل محاسبه را بنویسید)؟

تعداد ضرب:

در هر مکان ۶۴ تا feature map که هر کدام از کرنل ۳ در ۳ در ۳ استفاده کرده که با ضرب مولفه به مولفه نیاز به ۲۷ عمل ضرب دارد.

$$512 \times 512 \times (3 \times 3 \times 3) \times 64 = 452984832$$

تعداد جمع:

برای محاسبه تعداد عملیات جمع در یک مکان باید ۲۷ مولفه را با هم جمع کنیم که نیاز به ۲۶ عمل جمع است و یک عمل جمع هم برای بایاس استفاده شده است.

$$512 \times 512 \times (26 + 1) \times 64 = 452984832$$

پ) اگر بجای پیاده‌سازی کانوولوشنی، شبکه VGG16 را بر روی همان مکان‌هایی که در قسمت قبل اعمال شده است، به صورت عادی اعمال کنیم، تعداد ضرب و جمع لازم چه مقدار خواهد بود (در این قسمت تعداد ضرب و جمع در حالت عادی با ورودی ۲۲۴×۲۲۴ را محاسبه کنید و این عدد را در تعداد دفعاتی که لازم است شبکه اجرا شود ضرب کنید؟ نتیجه این قسمت را با قسمت قبل مقایسه کنید.

ابعاد خروجی قسمت قبل ۱۰ در ۱۰ بود یعنی عمل طبقه بندی را برای ۱۰۰ ناحیه محاسبه کرده است.

تعداد ضرب:

$$100 \times 224 \times 224 \times (3 \times 3 \times 3) \times 64 = 8670412800$$

تعداد جمع:

$$100 \times 224 \times 224 \times (26 + 1) \times 64 = 8670412800$$

$$\text{نسبت} = \frac{8670412800}{452984832} = 19$$

استفاده از شبکه FCN باعث ۱۹ برابر شدن سرعت سیستم می‌گردد.

VGG16 - Structural Details													
#	Input Image			output			Layer	Stride	Kernel		in	out	Param
1	224	224	3	224	224	64	conv3-64	1	3	3	3	64	1792
2	224	224	64	224	224	64	conv3064	1	3	3	64	64	36928
	224	224	64	112	112	64	maxpool	2	2	2	64	64	0
3	112	112	64	112	112	128	conv3-128	1	3	3	64	128	73856
4	112	112	128	112	112	128	conv3-128	1	3	3	128	128	147584
	112	112	128	56	56	128	maxpool	2	2	2	128	128	65664
5	56	56	128	56	56	256	conv3-256	1	3	3	128	256	295168
6	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
7	56	56	256	56	56	256	conv3-256	1	3	3	256	256	590080
	56	56	256	28	28	256	maxpool	2	2	2	256	256	0
8	28	28	256	28	28	512	conv3-512	1	3	3	256	512	1180160
9	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
10	28	28	512	28	28	512	conv3-512	1	3	3	512	512	2359808
	28	28	512	14	14	512	maxpool	2	2	2	512	512	0
11	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
12	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
13	14	14	512	14	14	512	conv3-512	1	3	3	512	512	2359808
	14	14	512	7	7	512	maxpool	2	2	2	512	512	0
14	1	1	25088	1	1	4096	fc		1	1	25088	4096	102764544
15	1	1	4096	1	1	4096	fc		1	1	4096	4096	16781312
16	1	1	4096	1	1	1000	fc		1	1	4096	1000	4097000
Total													**138,423,208**