دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

گزارش درس بینایی ماشین

تشخیص اشیاء با شبکه ترنسفورمر

نگارش

مینو دولت‌آبادی

استاد درس

دکتر رضا صفابخش

دی ۱۴۰۱

چکیده

اگرچه تلاش علم یادگیری ماشین، حداقل کردن دستورالعمل‌های صریح انسانی و یادگیری مستقیم از داده‌هاست؛ این هدف تا پیش از این در حوزه تشخیص اشیاء کمتر در دسترس بوده‌است. شبکه‌های رایج تشخیص اشیاء دارای بسیاری از اجزای از پیش طراحی شده مانند سرکوب غیرحداکثری و پیشنهاد ناحیه هستند. این اجزا که به نوعی دانش پیشین طراحان از مسئله تشخیص اشیاء را به شبکه تزریق می‌کنند، در طراحی‌های فعلی شبکه‌های تشخیص اشیاء اجتناب ناپذیر هستند. برای حل این مشکل، در سالهای اخیر رویکردی جدید برای تشخیص اشیاء بر مبنای شبکه ترنسفورمر ارائه شده‌است. در این رویکرد، مسئله تشخیص اشیاء با یک دیدگاه تازه، به عنوان یک مسئله پیش‌بینی مجموعه فرمول‌بندی شده‌است. این فرمول بندی جدید، نیاز شبکه را به اجزای از پیش طراحی شده از بین می‌برد اما دارای ضعف‌هایی است. برای حل این ضعف‌ها، که مهمترین آنها سرعت همگرایی پایین در آموزش شبکه است، کارهای زیادی ارائه شده که در این گزارش به آن‌ها می‌پردازیم.

واژه‌های کلیدی:

تشخیص اشیاء، شبکه ترنسفورمر، شبکه DETR، مکانیزم توجه، پیش آموزش بدون نظارت

فهرست مطالب

صفحه

فهرست اشکال

صفحه

فهرست جداول

صفحه

فصل اول

مقدمه

مقدمه

الگوریتم‌های تشخیص اشیاء[1] برای سال‌های طولانی شامل دو رویکرد اصلی دومرحله‌ای[2] و یک مرحله‌ای[3] بوده‌اند[۱]. ویژگی مشترک هر دو رویکرد، وابستگی آن‌ها به نوعی حدس اولیه درباره مکان اشیاء است که تاثیر به سزایی در عملکرد نهایی الگوریتم دارد[۲]. این حدس اولیه که به صورت اجزا و فرآیندهای از پیش طراحی شده[4] در مدل گنجانده می‌شود، در الگوریتم‌های دو مرحله در قالب پیشنهاد ناحیه[5][۳, ۴] و در الگوریتم‌های تک مرحله معمولا به صورت نقاط لنگرگاه[6] است[۵, ۶]. علاوه بر این، هر دو رویکرد دارای اجزای پیش طراحی شده دیگری به عنوان مثال برای حذف تشخیص‌های تکراری هستند که کل الگوریتم را تا حد زیادی به دانش پیشین مهندسان نسبت به مسئله وابسته می‌کند.

در حوزه پردازش زبان‌های طبیعی[7]، با معرفی شبکه‌های ترنسفورمر[8][۷] جریان انقلابی شروع شد که یکی از مهمترین نتایج آن، علاوه بر بهبود عملکرد در مسائل مختلف، کم رنگ شدن نیاز الگوریتم‌ها به اجزای از پیش طراحی شده و به بیان دیگر، یکپارچه شدن الگوریتم‌ها بود؛ به طوری که با یک مرحله پیش آموزش و کمترین تغییرات ممکن، شبکه برت[9][۸] توانست در یازده مسئله مختلف از حوزه پردازش زبان‌های طبیعی، بهترین نتایج را در مقایسه با سایر روش‌ها کسب کند.

با در نظر گرفتن همین موضوع، کاریون و همکاران[۹] یک شبکه کاملا یکپارچه برای تشخیص اشیاء معرفی کردند که برخلاف تلاش‌های پیشین، از نظر دقت، قابلیت رقابت با رویکرد دومرحله‌ای را داشت. الگوریتم پیشنهادی آن‌ها که بر اساس نسخه اصلی شبکه ترنسفورمر است، مسئله تشخیص اشیاء را به

---

[1] Object detection

[2] Two-stage

[3] Single-stage

[4] Handcrafted components/processes

[5] Region proposal

[6] Anchor points

[7] Natural language processing

[8] Transformer

[9] BERT

صورت یک مسئله پیش‌بینی مجموعه[10] فرموله می‌کند. بر این اساس، نوآوری اصلی آن‌ها یک تابع هزینه مبتنی بر تخمین مجموعه است که نیاز به اجزای از پیش طراحی شده را برای حذف تشخیص‌های تکراری از بین می‌برد.

الگوریتم پیشنهادی آن‌ها، DETR، اگرچه یک جهش بزرگ به سمت ساده‌سازی الگوریتم‌های تشخیص اشیاء است، اما از چند ضعف بزرگ رنج می‌برد:

- در مقایسه با الگوریتم‌های رایج، به تعداد تکرار بسیار بیشتری برای همگرایی نیاز دارد[۹].
- در دیتاست‌های کوچک نتایج ضعیف‌تری دارد[۱۰].
- با توجه به پیچیدگی بالای شبکه و عدم امکان استفاده از نقشه‌های ویژگی[11] با ابعاد بزرگ در آن، در تشخیص اشیاء کوچک عمکرد مناسبی ندارد[۱۱].

علارغم این مشکلات، چشم‌اندازه تازه شبکه DETR، محققان بسیاری را در سال‌های اخیر به حل این مشکلات علاقه‌مند کرده‌است. درهمین راستا، بسیاری از بهبودهایی که محققان برای حل این مشکلات ارائه کرده‌اند، شامل ایجاد تغییرات ساختاری در شبکه DETR بوده‌اند. ژو و همکاران[۱۱] با هدف بهبود دقت الگوریتم در تشخیص اشیاء کوچک و بهبود سرعت همگرایی آن، مکانیزم توجه[12] جدید، تحت عنوان مکانیزم توجه بی‌شکل[13] معرفی کردند که هزینه محاسباتی این مکانیزم را به شدت کاهش می‌دهد. در نتیجه، یادگیری شبکه سریعتر می‌شود. همچین امکان استفاده از نقشه‌های ویژگی با ابعاد بالاتر موجب شد که دقت آن در تشخیص اشیاء کوچک نیز بهبود یابد. با تزریق اطلاعات مکانی به ساختار شبکه DETR، لیو و همکاران[۱۲] از دانشهای پیشین موجود در مسئله تشخیص اشیاء برای افزایش سرعت همگرایی و دقت استفاده کردند.

اگرچه کارهای مختلف ثابت کرده‌اند که افزودن تغییرات ساختاری موجب بهبود عملکرد شبکه DETR می‌شود؛ اما این تغییرات ساختاری که معمولا شامل افزودن یک یا چند جز از پیش طراحی شده به شبکه

[10] Set prediction problem

[11] Feature map

[12] Attention

[13] Deformable

است، با هدف نهایی کاریون و همکاران[۹]، ساده‌سازی مسئله تشخیص اشیاء، سازگار نیست. به همین جهت برخی از محققین تلاش داشتند تا بدون تغییر در ساختار نهایی شبکه DETR، ضعف‌های آن را برطرف کنند. لی و همکاران[۱۳] با افزودن یک وظیفه کمکی رفع نویز به ترنسفورمر، فرآیند یادگیری شبکه را به خصوص در مراحل اولیه آموزش بهبود دادند. این وظیفه کمکی، شامل افزودن نویز به تشخیص‌های مبنا[۱۴] و تشویق شبکه برای رفع نویز از آن‌ها است. در نهایت، برای برطرف کردن ضعف شبکه DETR در دیتاست‌های کوچک، دای و همکاران[۱۰] از یک روش جدید پیش‌آموزش بدون نظارت، مشابه آنچه در حوزه پردازش زبان‌های طبیعی رایج است[۸]، استفاده کردند.

هدف از این گزارش، معرفی شبکه DETR [۹] و چند مورد از بهبودهای آن است. در فصل دوم، دیدگاه جدید ارائه‌دهندگان این شبکه را به صورت مفصل تشریح کرده و جزئیات شبکه DETR و اهمیت ساختار ترنسفومر در آن را بررسی می‌کنیم. در فصل سوم برخی از بهبودهای ساختاری شبکه DETR، شامل کارهای ژو و همکاران[۱۱] و لیو و همکاران[۱۲] را ارائه می‌کنیم. فصل چهارم به دو کار جهت بهبود شبکه DETR، بدون ایجاد تغییر در ساختار آن[۱۰, ۱۳] اختصاص دارد. در نهایت، در فصل پنجم، مروری بر کارهای معرفی شده و تاثیر شبکه DETR بر حوزه بینایی ماشین خواهیم داشت.

[14] Ground truth

فصل دوم

تشخیص اشیاء با شبکه ترنسفورمر

تشخیص اشیاء با شبکه ترنسفورمر

به نظر می‌رسد مهم‌ترین نوآوری کاریون و همکاران[۹] در الگوریتم پیشنهادی آن‌ها(DETR) نه استفاده از شبکه ترنسفورمر بلکه تعریف جدید آن‌ها از مسئله تشخیص اشیاء است. در این تعریف جدید، تشخیص اشیاء به صورت یک مسئله پیشبینی مجموعه فرموله می‌شود. یک خاصیت مهم مجموعه‌ها، تکراری نبودن اعضای آن‌ها است؛ بنابراین در صورت تعریف مناسب مسئله در این فرمول‌بندی جدید، نیاز به راهکاری برای حذف تشخیص‌های تکراری وجود نخواهد داشت. با درنظر گرفتن این موضوع، ابتدا این فرمول‌بندی جدید را تشریح می‌کنیم و سپس در ادامه فصل، جزئیات شبکه DETR و اهمیت استفاده از ترنسفورمر در آن را توضیح خواهیم داد.

۲- ۱- تشخیص اشیاء به عنوان یک مسئله تخمین مجموعه

فرض کنید y مجموعه اشیاء مبنای[۱۵] موجود در تصویر و $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ مجموعه N شی تشخیص داده شده توسط شبکه باشد(این مجموعه می‌تواند شامل کلاس پس‌زمینه باشد). با فرض اینکه N برای همه تصاویر ثابت و بسیار بزرگتر از مجموعه y باشد، مجموعه y به وسیله کلاس پس‌زمینه(∅) تا رسیدن به اندازه $\hat{y}$ گسترش می‌یابد. سپس با درنظر گرفتن $\mathcal{L}_{\text{match}}$ به عنوان تابع هزینه، یک تناظر یک به یک بین مجموعه مبنا و مجموعه پیش‌بینی بدست می‌آید:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg min} \sum_{i}^{N} \mathcal{L}_{\text{match}}\left(y_i, \hat{y}_{\sigma(i)}\right) \qquad 2-1$$

این تناظر دو بخشی[۱۶] بهینه، با هزینه محاسباتی معقولی توسط الگوریتم مجارستانی[۱۷] انجام می‌شود.

تابع هزینه مناسب باید به طور همزمان برچسب کلاس و کادرمحصورکننده اشیاء مبنا و پیش‌بینی شده را درنظر بگیرد. هر عضو از مجموعه مبنا، به صورت $y_i = (c_i, b_i)$ تعریف می‌شود که c_i برچسب کلاس

[15] Ground truth

[16] Bipartite

[17] Hungarian algorithm

و $b_i \in [0,1]^4$ برداری است که مختصات و ابعاد کادر محصورکننده را به صورت نرمال شده نسب به ابعاد تصویر مشخص می‌کند. برای شی پیش‌بینی شده با اندیس $\sigma(\mathrm{i})$ احتمال کلاس c_i به صورت $\hat{p}_{\sigma(i)}(c_i)$ و کادرمحصورکننده پیش‌بینی شده با $\hat{b}_{\sigma(i)}$ نمایش داده می‌شود. با این تعریف، کاریون و همکاران[۹] تابع هزینه تطابق را به صورت زیر درنظر گرفتند:

$$-\mathbb{1}_{\{c_i \neq \emptyset\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\sigma(i)}\right) \qquad 2-2$$

که $\mathcal{L}_{\text{box}}$ در ادامه همین بخش تعریف خواهد شد. بعد از تعیین تناظر یک به یک میان دو مجموعه y و $\hat{y}$، گام بعدی محاسبه کردن تابع هزینه برای آموزش شبکه است. این تابع هزینه نیز باید برچسب کلاس و کادرمحصورکننده را به صورت همزمان درنظر بگیرد:

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{\text{box}}\left(b_i, \hat{b}_{\hat{\sigma}}(i)\right) \right] \qquad 2-3$$

این تابع هزینه مانند سایر الگوریتم‌های رایج، ترکیبی از قرینه لگاریتم درست‌نمایی[۱۸] برای برچسب کلاس پیش‌بینی شده و یک تابع هزینه برای کادرمحصورکننده است که به صورت زیر تعریف می‌شود:

$$\mathcal{L}_{\text{box}} = \lambda_{\text{iou}} \mathcal{L}_{\text{iou}}\left(b_i, \hat{b}_{\sigma(i)}\right) + \lambda_{\text{L1}} \left\| b_i - \hat{b}_{\sigma(i)} \right\|_1 \qquad 2-4$$

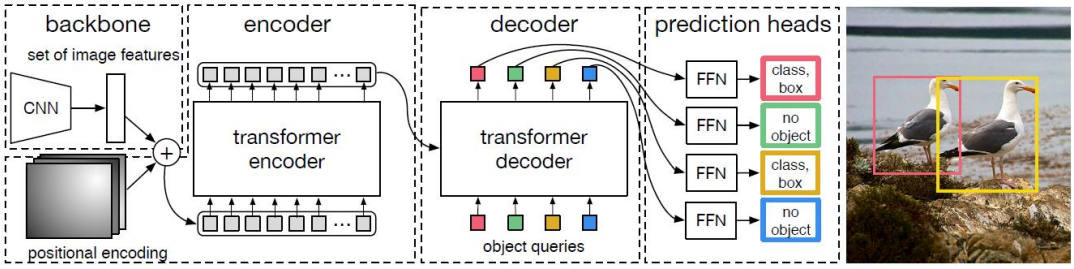
در رابطه فوق λ_{L1} و $\mathcal{L}_{\text{iou}}$ هایپرپارامترهای تنظیم هستند. جمله دوم معادله فوق، تابع هزینه رایج ℓ_1 است. از آنجایی که این تابع هزینه نسبت به ابعاد کادرمحصورکننده حساس است؛ با تابع هزینه IoU تعمیم یافته[۱۴] جمع شده‌است.

پس از تشریح فرمول‌بندی جدید که توسط کاریون و همکاران[۹] ابداع شد، در بخش بعد به الگوریتم کلی آن‌ها و اهمیت استفاده از شبکه ترنسفورمر در آن می‌پردازیم.

۲- ۲- ساختارکلی شبکه

تابع هزینه جدیدی که در بخش قبل برای مسئله تشخیص اشیاء معرفی شد ایجاب می‌کند که شبکه‌ای که برای تخمین مجموعه استفاده می‌شود، از طریق سازوکاری، ارتباطی بین اجزای

[18] Log-likelihood



شکل ۲-۱-ساختار شبکه DETR [۹]

مجموعه(پیش‌بینی‌های شبکه) برقرار کند. از این طریق هر عضو مجموعه از سایر اعضا آگاه بوده و شبکه تا حد ممکن به سمت تشخیص‌های یکتا حرکت می‌کند. نظر کاریون و همکاران[۹] این بود که مکانیزم خودتوجه[۱۹] در شبکه ترنسفورمر، این سازوکار را به خوبی پیاده‌سازی می‌کند و درنتیجه آن‌ها این شبکه را برای تشخیص اشیاء، با این دیدگاه جدید استفاده کردند.

ساختار کلی شبکه DETR در شکل ۲-۱ نمایش داده شده‌است. پس از استخراج ویژگی تصویر $x_{\text{img}} \in \mathbb{R}^{3\times H_0\times W_0}$ توسط یک شبکه شالوده[۲۰] کانولوشنی، ویژگی‌های استخراج شده $f \in \mathbb{R}^{C\times\frac{H_0}{32}\times\frac{W_0}{32}}$ با استفاده از یک تبدیل خطی به ابعاد کوچکتر $z_0 \in \mathbb{R}^{d\times\frac{H_0}{32}\times\frac{W_0}{32}}$ نگاشت می‌شوند. مشابه شبکه استاندارد ترنسفورمر[۷] این ویژگی‌ها پس از ورود به بخش کدگذار[۲۱] شبکه ترنسفورمر، پیش از هر لایه خودتوجه با انکودینگ مکانی[۲۲] جمع می‌شوند. پس از استخراج ویژگی‌های سطح بالاتر توسط کدگذار، این ویژگی‌ها به بخش کدگشا[۲۳] ارسال می‌شود.

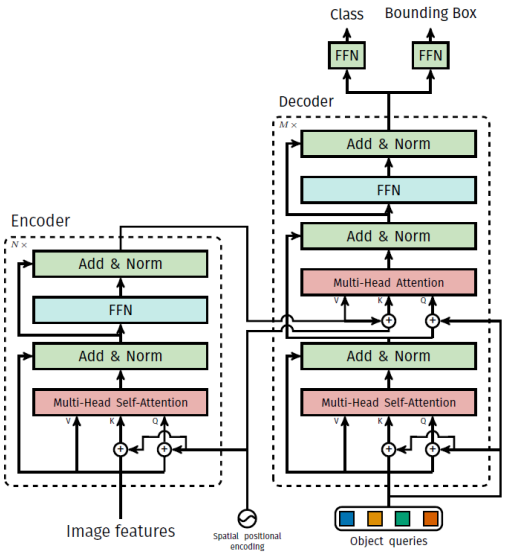
[19] Self-Attention

[20] Backbone

[21] Encoder

[22] Positional encodings

[23] Decoder



شکل ۲-۲-جزئیات شبکه DETR [۹]

ورودی بخش کدگشا، N توکن امبدینگ مکانی است که در طول آموزش شبکه یادگرفته می‌شوند و به آن‌ها جستارهای اشیاء[۲۴] گفته می‌شود. مشابه امبدینگ‌های مکانی بخش کدگذار، این جستارها در ورودی هر لایه خودتوجه از کدگشا، با خروجی لایه‌های قبل جمع می‌شود. شکل ۲-۲ ساختار تشریح شده را با جزئیات کامل نشان می‌دهد. در نهایت خروجی بخش کدگشا به وسیله دو شبکه جلورو[۲۵] مشترک برای همه N جستار، به احتمالات کلاس‌ها و کادرمحصورکننده تبدیل می‌شود.

طی آزمایشات، کاریون و همکاران[۹] مشاهده کردند که محاسبه خروجی‌ها و سپس تابع هزینه پس از هر لایه از بخش کدگشا، به بهبود عملکرد شبکه کمک می‌کند. در این حالت شبکه‌های جلورو برای تمامی لایه‌ها مشترک خواهد بود.

[24] Object queries

[25] Feed-forward

جدول ۲-۱-نتایج شبکه DETR در مقایسه با شبکه Faster-RCNN [۹]

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

۲- ۳- نتایج اولیه

جدول ۲-۱ نتایج شبکه DETR را در مقایسه با شبکه مطرح Faster-RCNN [۴] در دیتاست کوکو[۲۶][۱۵] نمایش می‌دهد. علامت + در بخش میانی نمایانگر آموزش طولانی‌تر(۹ برابر نسب به حالت استاندارد) برای شبکه Faster-RCNN است. نتایج نسخه‌های مختلف DETR نیز فقط برای همین آموزش طولانی ارائه شده است. با توجه به این نتایج، با تعداد پارامترهای برابر، شبکه DETR به صورت کلی عملکرد بهتری داشته؛ اما برای اشیاء با اندازه کوچک (AP_S) عملکرد ضعیف‌تری از خود نشان داده‌است. این ضعف، با توجه به اینکه رزلوشن ورودی این شبکه بسیار کوچکتر از شبکه Faster-RCNN است، دور از انتظار نبود.

نقطه ضعف مهم‌تری که نویسندگان مقاله کمتر به آن پرداخته‌اند، میزان آموزش موردنیاز شبکه DETR برای رسیدن به این نتایج است. همانطور که در جدول ۲-۱ مشاهده کردید، شبکه Faster-RCNN با زمان آموزش حدودا یک دهم (بخش اول از جدول)، به نتایج نسبتا خوبی رسیده‌است. متاسفانه در مقاله فعلی نتایج آموزش کوتاه‌تر برای شبکه DETR ارائه نشده‌است؛ اما در فصل بعد بیشتر به این موضوع خواهیم پرداخت.

[26] COCO

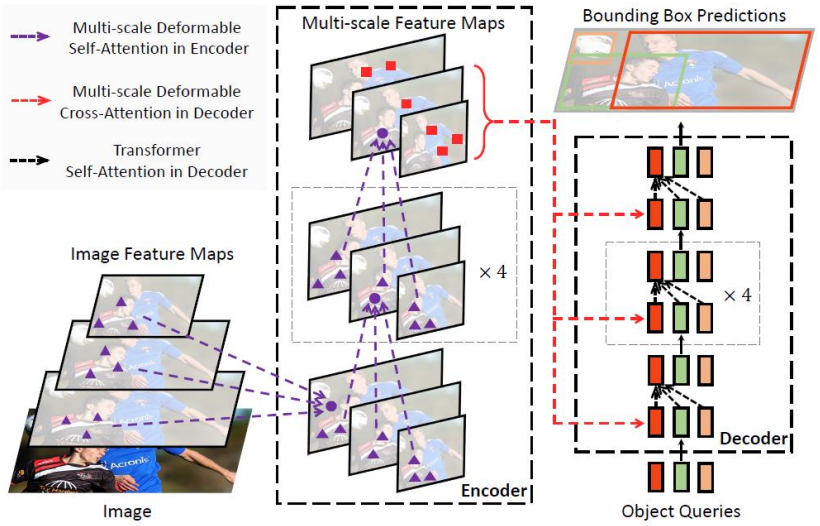
فصل سوم

بهبودهای ساختاری

بهبودهای ساختاری

مهم‌ترین ضعف‌های شبکه DETR، سرعت همگرایی پایین و عملکرد ضعیف در تشخیص اشیاء کوچک، تمرکز اصلی محققین این حوزه را به خود جلب کرده‌است. از بین این دو ضعف، مشکل شبکه در تشخیص اشیاء کوچک توجیه ظاهراً ساده‌ای دارد: نقشه‌های ویژگی‌های ورودی به این شبکه، ابعاد بسیار کوچکی دارند(۱/۳۲ ابعاد تصویر اصلی). علارغم ظاهر ساده این مشکل، تلاش برای حل آن به مشکل بسیار بزرگتری منجر می‌شود. با توجه به پیچیدگی بالای مکانیزم خودتوجه در شبکه ترنسفورمر، افزایش حتی دو برابری ابعاد نقشه‌های ویژگی، موجب شانزده برابر شدن محاسبات مکانیزم خودتوجه بخش کدگذار و افزایش تقریبا دوبرابری بار محاسباتی کل شبکه می‌شود[۹]. بنابراین به نظر می‌رسد راهی به جز تغییر ساختاری در شبکه برای کاهش این پیچیدگی وجود ندارد.

یافتن علت ضعف دیگر شبکه DETR، سرعت همگرایی پایین، نیاز به آزمایشات بیشتری دارد. در حقیقت مقالات مختلف، فرضیات مختلفی برای علت این ضعف ارائه کرده‌اند که به نتایج مختلفی منجر شده‌است. از میان این فرضیات، دو موردی که منجر به ایجاد تغییرات ساختاری در شبکه شده‌اند را در این فصل بررسی می‌کنیم.



شکل ۳-۱-ساختار شبکه Deformable-DETR [۱۱]

۳- ۱- استفاده از مکانیزم توجه بی‌شکل[1]

شبکه‌های رایج تشخیص اشیاء، معمولا از نقشه‌های ویژگی با مقیاس‌های مختلف به عنوان ورودی استفاده می‌کنند. استفاده از نقشه‌های ویژگی با ابعاد بزرگتر، توانایی شبکه در یافتن اشیاء کوچک را افزایش می‌دهد. حال آن که در شبکه DETR به دلیل بار محاسباتی بالای مکانیزم توجه، استفاده از نقشه‌های ویژگی بزرگتر عملی نیست. برای حل این مشکل، ژو و همکاران[۱۱] شبکه Deformable-DETR را معرفی کردند. آن‌ها با معرفی لایه‌های توجه بی‌شکل، بار محاسباتی مکانیزم توجه در شبکه ترنسفورمر را کاهش دادند و از این طریق، ضمن ایجاد قابلیت استفاده از نقشه‌های ویژگی بزرگتر، سرعت همگرایی شبکه را نیز افزایش دادند. ساختار شبکه پیشنهادی آن‌ها کاملا مشابه شبکه DETR است. با این تفاوت که ورودی‌های این شبکه، نقشه‌های ویژگی با مقیاس‌های مختلف هستند(شکل ۳-۱) و مکانیزم خودتوجه در بخش کدگذار و مکانیزم توجه کدگذار-کدگشا در بخش کدگشا با مکانیزم توجه بی‌شکل جایگزین شده‌اند که در بخش بعد آن را توضیح خواهیم داد.

[1] Deformable attention

۳- ۱- ۱- مکانیزم توجه و مکانیزم توجه بی شکل

برای درک مکانیزم توجه بی‌شکل، ابتدا به معرفی ساختار اصلی مکانیزم توجه می‌پردازیم. فرض کنید $q \in \Omega_q$ اندیس یک جستار با بردار ویژگی $\boldsymbol{z}_q \in \mathbb{R}^C$ و $k \in \Omega_k$ اندیس یک کلید[1] با بردار ویژگی $\boldsymbol{x}_k \in \mathbb{R}^C$ باشد. مکانیزم توجه چندواحدی[2] به صورت زیر تعریف می‌شود[۷]:

$$\text{MultiHeadAttn}(\boldsymbol{z}_q, \boldsymbol{x}) = \sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{k \in \Omega_k} A_{mqk} \cdot \boldsymbol{W}'_m \boldsymbol{x}_k \right] \qquad 3-1$$

که m اندیس واحدهای توجه و $W'_m \in \mathbb{R}^{C_v \times C}$ و $W_m \in \mathbb{R}^{C \times C_v}$ وزن‌های یادگرفتنی هستند. وزن‌های توجه $A_{mqk} \propto \exp\left\{\frac{\boldsymbol{z}_q^1 \boldsymbol{U}_m^T \boldsymbol{V}_m \boldsymbol{x}_k}{\sqrt{C_m}}\right\}$ به گونه‌ای نرمالسازی می‌شوند که جمع آن‌ها صفر باشد. همچنین $\boldsymbol{U}_m, \boldsymbol{V}_m \in \mathbb{R}^{C_v \times C}$ نیز وزن‌های یادگرفتنی هستند.

ژو و همکاران[۱۱]، تعریف مکانیزم توجه به صورت بالا را عامل اصلی مشکلات شبکه DETR می‌دانند. از یک جهت، با افزایش تعداد کلیدها، توزیع ماتریس وزن‌های توجه A_{mqk} به سمت توزیع یکنواخت میل می‌کند[۱۱, ۱۲] که خود فرآیند یادگیری را طولانی‌تر می‌کند. از جهت دیگر، افزایش تعداد کلیدها، که در مورد شبکه DETR به اندازه ابعاد نقشه ویژگی‌است، حجم محاسبات شبکه را به شدت افزایش می‌دهد (جدول ۳-۱).

ژو و همکاران[۱۱] برای حل مشکلات فوق، با الهام از شبکه‌های کانولوشنی بی‌شکل[۱۶]، یک مکانیزم توجه بی شکل معرفی و سپس آن را به مکانیزم توجه بی‌شکل چندمقیاسی[3] توسعه دادند.

جدول ۳-۱-مقایسه پیچیدگی محاسباتی مکانیزم توجه عادی و مکانیزم توجه بی‌شکل [۱۱]

Standard Attention	$O(H^2W^2C)$	$O(HWC^2 + NHWC)$
Deformable Attention	$O(HWC^2)$	$O(NKC^2)$

HW: spatioal size, C: transormer embeding dimention, N: number of object queries, K: total sampled key

[1] Key

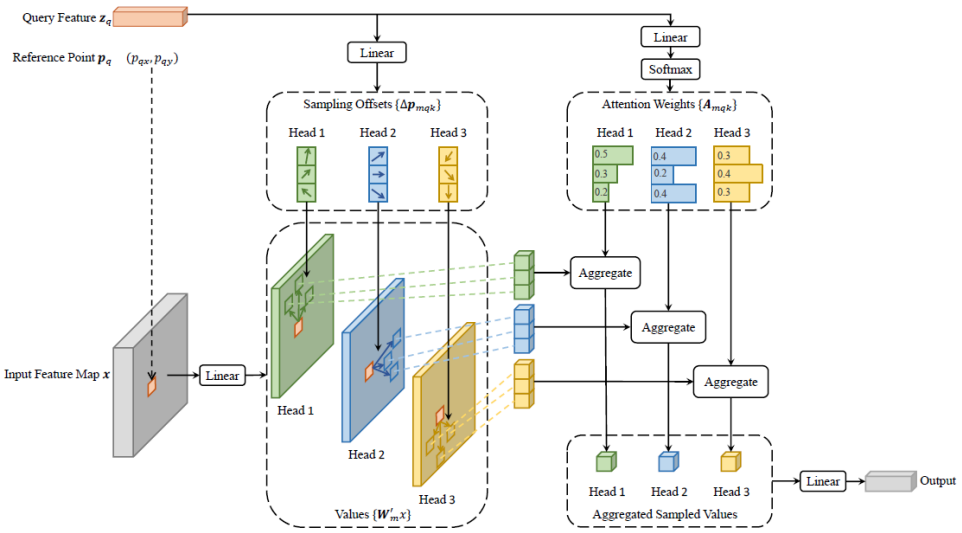
[2] Multi-head attention

[3] Multi-scale Deformable Attention

فرض کنید که $\{\boldsymbol{x}^l\}_{l=1}^{L}$ نقشه‌های ویژگی چند مقیاسی ورودی هستند که $\boldsymbol{x}^l \in \mathbb{R}^{C\times H_l\times W_l}$. نقطه $\hat{\boldsymbol{p}}_q \in [0,1]^2$ یک نقطه مرجع برای هر کلید q است که مختصات آن در نقشه‌های ویژگی به صورت نرمال‌شده در نظر گرفته شده است. مکانیزم توجه بی‌شکل چندمقیاسی به صورت زیر تعریف می‌شود:

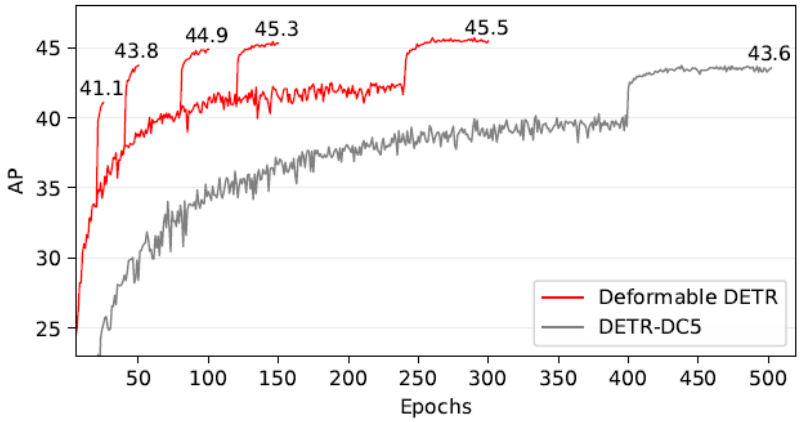
$$\text{MSDeformAttn}\left(\boldsymbol{z}_q, \hat{\boldsymbol{p}}_q, \{\boldsymbol{x}^l\}_{l=1}^{L}\right) = \sum_{m=1}^{M} \boldsymbol{W}_m \left[\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk}\cdot \boldsymbol{W}'_m \boldsymbol{x}^l\left(\phi_l\left(\hat{\boldsymbol{p}}_q\right) + \Delta\boldsymbol{p}_{mlqk}\right)\right] \qquad 3-2$$

در رابطه بالا K تعداد نقاط نمونه‌برداری (K<<HW)، $\Delta\boldsymbol{p}_{mlqk}$ آفست نمونه‌برداری[1] و A_{mlqk} که $\sum_{l=1}^{L}\sum_{k=1}^{K} A_{mlqk} = 1$ وزن‌های توجه هستند. وزن‌های توجه و آفست نمونه‌برداری با یک تبدیل خطی روی ویژگی‌های جستار $\boldsymbol{z}_q$ بدست می‌آیند. به صورت شهودی، مکانیزم توجه بی‌شکل، به جای توجه به تمام کلیدها، به تعداد کمی کلید (۴ کلید) که از اطراف یک نقطه مرجع، برداشت می‌شوند، توجه می‌کند. نقطه نمونه برداری برای مکانیزم خود-توجه که در آن جستارها و کلیدها یکسان هستند، مختصات خود جستار است(شکل ۳-۲) و برای مکانیزم توجه کدگذار-کدگشا، با یک تبدیل خطی روی



شکل ۳-۲-مکانیزم خودتوجه بی‌شکل [۱۱]

[1] sampling offset



شکل ۳-۳-مقایسه سرعت همگرایی شبکه DETR و Deformable-DETR [۱۱]

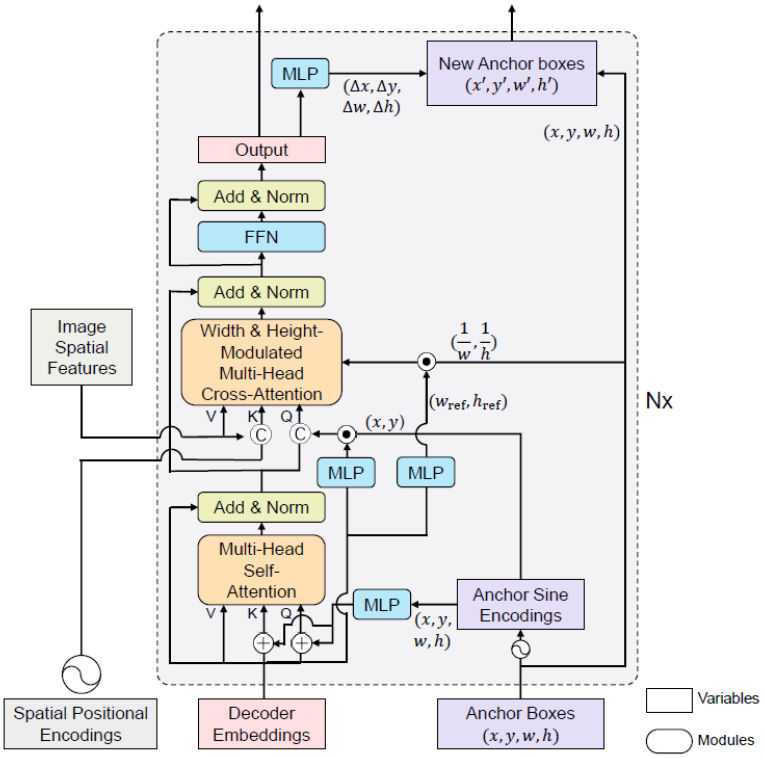
جستار بدست می‌آید. از آنجایی که ویژگی‌های مربوط به این جستار، از اطراف نقطه کلیدی برداشت می‌شود، ژو و همکاران[۱۱] پیشنهاد دادند که مختصات کادرهای محصورکننده که توسط شبکه‌های جلورو پیش‌بینی می‌شود نیز نسبت به همین نقاط کلیدی پیش‌بینی شوند. آزمایشات آن‌ها نشان داد این کار آموزش شبکه را بیش از پیش تسهیل می‌کند.

۳- ۱- ۲- نتایج

فرض ژو و همکاران[۱۱] در مورد سرعت همگرایی پایین شبکه DETR این بود که پیچیدگی بالای مکانیزم توجه در ترنسفورمر عامل آن است. بر اساس همین فرض و با تلاش برای استفاده از ویژگی‌های چندمقیاسی، شبکه Deformable-DETR را پیشنهاد دادند. در شکل ۳-۳ سرعت همگرایی و دقت نهایی این شبکه در مقایسه با شبکه DETR نمایش داده شده است. در این شکل هر نمودار قرمز رنگ، آموزش شبکه Deformable-DETR را به ازای برنامه‌های مختلف کاهش ضریب یادگیری نشان می‌دهد که در همه آن‌ها این شبکه بسیار زودتر از شبکه DETR همگرا شده است. همچنین استفاده از ویژگی‌های چندمقیاسی به عنوان ورودی این شبکه، دقت تشخیص اشیاء کوچک که ضعف دیگر شبکه DETR بود تا حد شبکه Faster-RCNN بهبود داد.

۳- ۲- اطلاعات مکانی به عنوان دانش پیشین

کارهای مختلف بعد از معرفی شبکه DETR تقریبا بر روی این فرض هم عقیده هستند که عامل اصلی کند بودن همگرایی شبکه‌های تشخیص اشیاء مبتنی بر ترنسفورمر، بخش کدگشای آن و به ویژه مکانیزم توجه کدگذار-کدگشا است[۱۷]. مقایسه این مکانیزم با مکانیزم خود-توجه بخش کدگذار، نشان می‌دهد که تفاوت اصلی آن‌ها، وجود جستارهای اشیاء در مکانیزم توجه کدگذار-کدگشا، احتمالا ریشه اصلی این مشکل است. لیو و همکاران[۱۲] پیشنهاد دادند که افزودن اطلاعات مکانی به شبکه و به خصوص جستارهای اشیاء، احتمالا بتواند آموزش شبکه را بهبود دهد. اگرچه این کار به نوعی افزودن ماژول‌های از پیش طراحی شده به شبکه DETR است که با اهداف اصلی پیشنهاد دهندگان آن در تضاد است؛ اما نتایج آزمایشات حاکی از آن است که این تغییر سرعت همگرایی و دقت شبکه را به شدت افزایش می‌دهد.



شکل ۳-۴- ساختار بخش کدگشای شبکه DAB-DETR [۱۲]

۳- ۲- ۱- جستارهای اشیاء آگاه از مکان

ایده شـــبکه پیشـــنهادی لیو و همکاران[۱۲]، DAB-DETR، با جایگزین کردن جســـتارهای اشـــیاء با بردارهای چهاربعدی مختصات (x,y,w,h) شروع می‌شود. در حقیقت آن‌ها جستارهای اشیاء را صریحا به صورت مختصات کادرهای محصورکننده فرموله کردند که در هر لایه از بخش کدگشا، اصلاح می‌شوند. انجام این تغییر همانطور که در شکل ۳-۴ مشاهده می‌کنید، منجر به تغییرات غیر قابل اجتناب در شبکه می‌شود. به عنوان مثال، برای افزودن جستارهای اشیاء به عنوان انکودینگ مکانی، نیاز است که با استفاده از تبدیل غیرخطی آن‌ها با به ابعاد بالاتر منتقل کرد. علاوه بر این، اطلاعات مکانی موجود در اشـــیاء به صورت‌های مختلف به ساختارهای موجود در بخش کدگشا تزریق می‌شود. ذکر تمام جزئیات روابط جدید خارج از اهداف این گزارش است، اما تغییرات اصلی شبکه در شکل ۳-۴ قابل مشاهده‌است.

۳- ۲- ۲- نتایج

با توجه به جدول ۳-۲، ایده افزودن اطلاعات فضایی و درنظر گرفتن جستارهای اشیاء به عنوان مختصات کادرهای محصورکننده، سرعت همگرایی و دقت شبکه DAB-DETR نسبت به نسخه‌های دیگر شبکه DETR و شـــبکه Faster-RCNN بهبود می‌دهد. مشـــکلی که همچنان در این شـــبکه وجود دارد، دقت پایین آن در تشخیص اشیاء کوچک است که به نظر می‌رسد راهی به جز افزایش ابعاد نقشه‌های ویژگی یا استفاده از مکانیزم‌های توجه ساده شده مانند شبکه Deformable-DETR [۱۱] ندارد.

جدول ۳-۲-نتایج شبکه DAB-DETR [۱۲]

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
DETR-R50		500	42.0	62.4	44.2	20.5	45.8	61.1	86	41M
Faster RCNN-FPN-R50		108	42.0	62.1	45.5	26.6	45.5	53.4	180	42M
Anchor DETR-R50*		50	42.1	63.1	44.9	22.3	46.2	60.0	–	39M
Conditional DETR-R50		50	40.9	61.8	43.3	20.8	44.6	59.2	90	44M
DAB-DETR-R50		50	42.2	63.1	44.7	21.5	45.7	60.3	94	44M
DAB-DETR-R50*		50	**42.6**	63.2	45.6	21.8	46.2	61.1	100	44M

فصل چهارم

بهبودهای غیرساختاری

بهبودهای غیرساختاری

در فصل قبل دیدیم که برای رفع مشکل سرعت همگرایی شبکه DETR، مقالات مختلفی سعی در ایجاد تغییرات ساختاری در آن داشتند. همانطور که دیدیم این تغییرات باعث بهبودهای بسیار زیادی در این شبکه شده‌اند. علاوه بر این، دیدیم که برای برطرف نمودن ضعف این شبکه در تشخیص اشیاء کوچک، به نظر می‌رسد راهی جز بهینه کردن و کاهش پیچیدگی مکانیزم توجه در شبکه ترنسفومر وجود ندارد.

در این فصل به نوع دیگری از بهبودها، یعنی بهودهای غیرساختاری می‌پردازیم. منظور از بهبودهای غیرساختاری، تلاشهایی برای بهبود نتایج شبکه DETR بدون تغییر در ساختار اصلی آن است؛ هرچند این بهبودها ممکن است شامل افزودن بخش‌هایی به شبکه، تنها در مرحله آموزش آن باشد. به طور کلی بهبودهای غیرساختاری را می‌توان به دو دسته اصلی تقسیم کرد: بهبود با تعریف وظایف کمکی[1] در مرحله آموزش و بهبود از طریق پیش‌آموزش بدون نظارت[2]. مزیت اصلی این بهبودها این است که می‌توان آن‌ها را به سادگی به نسخه‌های پیشرفته‌تر DETR اضافه و از این طریق عملکرد شبکه را بیش از پیش بهبود داد.

۴- ۱- رفع نویز[3] به عنوان وظیفه کمکی

کارهای پیشین، جستارهای اشیاء یا مکانیزم توجه کدگذار-کدگشا را مسئول اصلی سرعت همگرایی پایین شبکه DETR می‌دانستند. اما یک جنبه از این شبکه که کمتر مورد بررسی قرار گرفته است، فرآیند تعیین تناظر بین تشخیص‌های مبنا و پیش‌بینی‌های شبکه با الگوریتم مجارستانی است. همانطور که در فصل دوم دیدیم این فرآیند(در این فصل به طور خلاصه آن را تطبیق مجارستانی می‌نامیم) که برگرفته از فرمول‌بندی جدید تشخیص اشیاء توسط کاریون و همکاران[۹] است، تمایز اصلی شبکه DETR با سایر الگوریتم‌های تشخیص اشیاء است؛ بنابراین یکی از دلایل سرعت همگرایی پایین، می‌تواند همین تمایز باشد. در همین راستا، آزمایشات لی و همکاران[۱۳] نشان داد که فرآیند تطبیق مجارستانی

[1] Auxiliary task

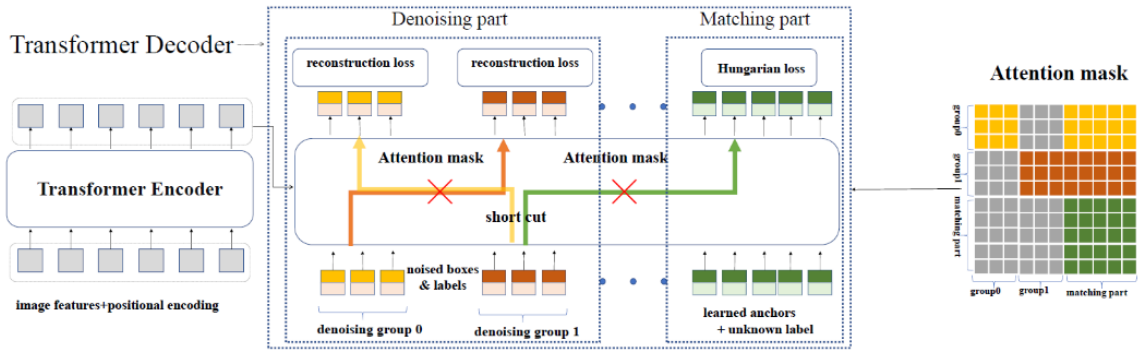
[2] Unsupervised Pre-training

[3] Denoising

به خصـــوص در گام‌های اولیه آموزش بســـیار ناپایدار بوده و با تغییرات ناگهانی در تابع هزینه، فرآیند یادگیری را دشـوار می‌کند. برای حل این مشـکل آن‌ها در حین آموزش، یک وظیفه کمکی رفع نویز برای شـبکه تعریف کردند. آن‌ها برای انجام این کار، علاوه بر جسـتارهای اشـیاء عادی که در شـبکه DETR وجود داشت، یک نسخه از تشخیص‌های مبنا را پس از اعمال نویز به عنوان جستارهای اضافه(نویزی) به ورودی کدگشـــا اعمال کردند. در این حالت، شـــبکه علاوه بر اینکه باید طبق معمول اشـــیاء موجود در تصویر را تشخیص دهد، باید تلاش کند که این جستارهای نویزی را به تشخیص‌های مبنای اصلی تبدیل کند. استدلال آن‌ها این بود، که این وظیفه کمکی که یادگیری آن برای شبکه آسان‌تر است، می‌تواند در حین آموزش، ضـمن راهنمایی شـبکه برای جسـتوجوی محلی به جای جسـتوجوی سـراسـری(از آنجا که پیش‌بینی تشـخیص‌های مبنا از روی تشـخیص‌های نویزی، بیشـتر یک جسـتوجوی محلی اسـت)، باعث جلوگیری از همپوشانی بین پیش‌بینی‌ها شده و آموزش شبکه را پایدارتر می‌کند.

۴-۱-۱- رفع نویز و ماسک توجه

در شـبکه جدید DN-DETR، جسـتارهای نویزی به P گروه $\mathbf{q} = \{\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{P-1}\}$ تقسـیم می‌شـوند که هر گروه شامل تمام تشخیص‌های مبنا هستند که به صورت تصادفی نویزی شده‌اند. منظور از نویزی کردن تشـــخیص‌های مبنا، تغییرهای کوچک در ابعاد و محل کادرهای محصـــورکننده و همچنین تغییر برچسـب کلاس‌ها به صـورت تصـادفی اسـت. شـبکه باید ضـمن انجام وظیفه معمول خود(تشـخیص اشـیاء براسـاس جسـتارهای اشـیاء)، این جسـتارهای نویزی را به نسـخه بدون نویز خود تبدیل کند. از آنجا که در



شکل ۴-۱-ساختار شبکه DN-DETR [۱۳]

این وظیفه جدید، تناظر بین پیش‌بینی شبکه و خروجی مورد انتظار ثابت است، تابع هزینه آن پایدارتر از تطبیق مجارستانی است.

افزودن جستارهای نویزی نیاز به ایجاد یک تغییر کوچک در مکانیزم خود-توجه بخش کدگشا دارد. در حلت عادی در این مکانیزم، تمامی جستارها می‌توانند با هم تبادل اطلاعات کنند. این تبادل اطلاعات بین جستارهای اشیاء و جستارهای نویزی به وضوح مطلوب نیست، زیرا ممکن است جستارهای اشیاء با نگاه کردن به جستارهای نویزی به جای تصویر، سعی در پیش‌بینی تشخیص‌های مبنا کنند. علاوه بر این، جستارهای نویزی از گروه‌های مختلف نیز به همین دلیل نباید امکان برقراری ارتباط داشته باشند. راهکار لی و همکاران[۱۳] با الهام گرفتن از ماسک توجه در نسخه استاندارد ترنسفورمر[۷]، تبدیل وزن‌ها توجه متناظر با ارتباطهای ممنوعه به منفی بی‌نهایت است. شکل ۴-۱ این ارتباط‌های ممنوعه و ماسک توجه مربوطه را نشان می‌دهد. در ماسک توجه‌ای که در سمت راست این شکل مشاهده می‌کنید، مربع‌های خاکستری نشان‌دهنده منفی بی‌نهایت است.

۴- ۱- ۲- نتایج

نقطه قوت روش پیشنهادی لی و همکاران[۱۳] این است که به سادگی می‌توان آن‌را به تمامی مدل‌های مبتنی بر DETR و حتی مدل‌های قدیمی‌تر اضافه کرد و آن‌ها را بهبود داد(جدول -۱۴). اما مدل اصلی آن‌ها DN-DETR، یک نسخه بهبود یافته از شبکه DAB-DETR است که در فصل قبل معرفی شد.

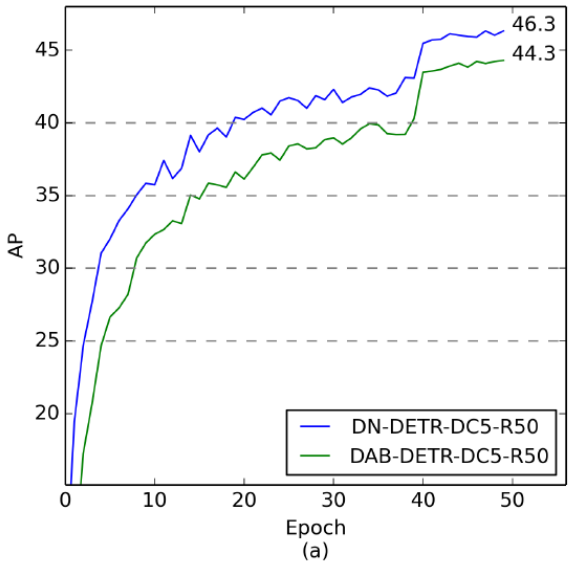
جدول ۴-۱-نتیجه اضافه کردن وظیفه کمکی رفع نویز به شبکه‌های مختلف [۱۳]

Model	MultiScale	#epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GFLOPs	Params
Extending DN to other detection models										
Anchor-DETR-DC5-R50 [21]		12	38.2	58.6	40.6	20.3	41.9	53.1	–	37M
DN-Anchor-DETR-DC5-R50		12	**39.4(+1.2)**	59.1	41.8	19.6	43.4	56.0	–	37M
Group-DAB-DETR-DC5-R50 [3]		12	41.9	–	–	23.3	45.6	58.4	–	–M
DN-Group-DAB-DETR-DC5-R50* [3]		12	**44.5(+2.6)**	–	–	25.9	48.2	62.2	–	–M
Faster R-CNN-FPN-R50 [21]	✓	12	37.9	58.8	41.1	22.4	41.1	49.1	180	40M
DN-Faster R-CNN-FPN-R50	✓	12	**38.4(+0.5)**	59.1	41.5	22.7	41.6	50.4	180	40M
SAM-DETR++-R50 [23]	✓	12	43.2	61.5	46.5	25.5	46.5	58.6	203	55M
DN-SAM-DETR++-R50* [23]	✓	12	**44.8(+1.6)**	62.6	47.9	26.7	48.2	60.9	203	55M
DINO-R50 w/o DN [24]	✓	12	46.0	64.0	49.9	29.3	49.2	60.5	279	47M
DINO-R50 w/ DN* [24]	✓	12	**47.4(+1.4)**	64.6	51.3	30.0	50.7	61.8	279	47M
Vanilla-DETR-R50 [1]		300	40.6	61.6	–	19.9	44.3	60.2	86	41M
DN-Vanilla-DETR-R50		300	**42.6(+2.0)**	62.3	44.9	21.6	46.1	61.4	86	37M
Extending DN to segmentation models										
Mask DINO-R50 w/o mask DN [11]	✓	12	40.7	62.8	43.7	21.0	43.4	60.6	234	50M
Mask DINO-R50 w/ mask DN * [11]	✓	12	**41.4(+0.7)**	62.9	44.6	21.1	44.2	61.4	234	50M
Mask2Former-R50 [5]	✓	12	38.7	59.8	41.2	18.2	41.5	59.8	226	44M
DN-Mask2Former-R50	✓	12	**39.7(+1.0)**	60.8	42.3	19.1	42.7	61.2	226	44M

همانطور که در شکل ۴-۲ مشاهده می‌کنید. شبکه DN-DETR از نظر دقت و سرعت همگرایی بهبود نسبتا مناسبی نسب به شبکه DAB-DETR دارد.

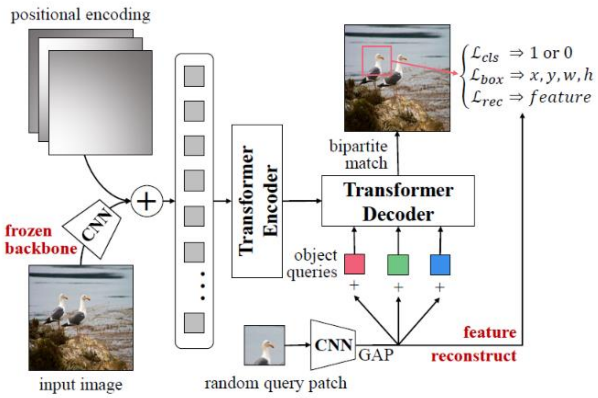
۴- ۲- پیش آموزش بدون نظارت

یکی از مشکلات شبکه‌های عصبی با تعداد پارامترهای زیاد، تعمیم پذیری ضعیف آن‌ها پس از آموزش در دیتاست‌های کوچک است. با استفاده از روش‌های پیش آموزش بدون نظارت، می‌توان از داده‌های بدون برچسب برای پیش آموزش شبکه و سپس انتقال یادگیری[1] برای دیتاست‌های کوچک‌تر استفاده کرد. پیش‌آموزش بدون نظارت همواره یک مفهوم جدا نشدنی از شبکه ترنسفورمر بوده‌است. در حقیقت شبکه ترنسفورمر بخش زیادی از محبوبیت خود در حوزه پردازش زبان‌های طبیعی را مدیون این مفهوم بوده‌است[۸]. موضوع پیش آموزش بدون نظارت در شبکه‌های مبتنی بر DETR توسط دای و همکاران[۱۰] مورد بررسی قرار گرفت. انگیزه آن‌ها برای پرداختن به این موضوع، مشاهده آن‌ها از نتایج



شکل ۴-۲-مقایسه شبکه DAB-DETR و DN-DETR [۱۳]

[1] Transfer learning



شکل ۴-۳-فرآیند پیش آموزش UP-DETR [۱۰]

ضعیف این شبکه در دیتاست پاسکال [۱۸](که تعداد داده‌های آموزشی نسبتا کمی دارد) بود. در ادامه روش پیشنهادی آن‌ها که مانند اکثر روش‌های پیش آموزش بدون نظارت، به سادگی قابل پیاده سازی است را تشریح می‌کنیم.

۴- ۲- ۱- تشخیص پچ‌های[1] تصادفی از تصویر

پیش آموزش به روش پیشنهادی دای و همکاران[۱۰]، UP-DETR ایده نسبتا ساده‌ای دارد. یک پچ از تصویر ورودی برش داده شده و پس از استخراج ویژگی با شبکه شالوده، به همه جستارهای اشیاء اضافه می‌شود (شکل۴-۳). وظیفه شبکه در این حالت پیش‌بینی محل دقیق پچ در تصویر است. بنابراین برای هر جستار شی در خروجی، چهار پارامتر مربوط به کادر محصورکننده و یک برچسب کلاس خواهیم داشت. برچسب کلاس مشخص می‌کند که آیا کادرمحصورکننده پیش‌بینی شده با پچ ورودی تطبیق دارد یا خیر. همانطور که مشاهده می‌کنید در اینجا مسئله دسته‌بندی، یک کلاسه است؛ در نتیجه در حین پیش آموزش، شبکه ممکن است ویژگی‌هایی که برای دسته‌بندی چند کلاسه نیاز است را به تدریج حذف کند. برای اجتناب از این اتفاق، دو ابتکار ساده در نظر گرفته شده‌است.

الف) برای اینکه جریان گرادیان مسئله یک کلاسه، خاصیت تمایزگری شبکه شالوده (که پیش آموزش دیده) را از بین نبرد، وزن‌های آن ثابت شده‌اند.

[1] patches

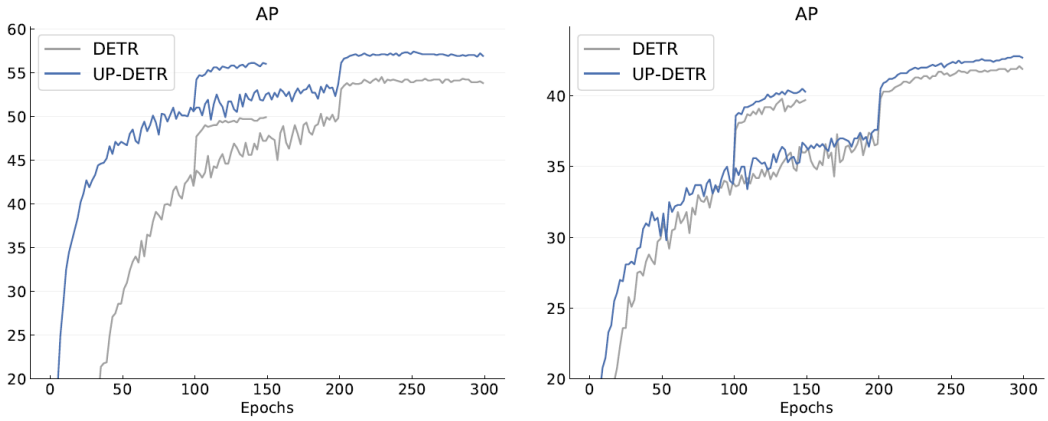
ب) برای اینکه این تمایزگری شبکه کانولوشنی، در طول شبکه ترنسفورمر نیز حفظ شود، یک وظیفه بازسازی جدید برای ترنسفورمر تعریف می‌شود. مطابق با این وظیفه، ترنسفورمر باید بتواند ویژگی‌های پچ که توسط شبکه شالوده استخراج شده را بازسازی کند. بدین منظور از تابع هزینه نرمال شده ℓ_2 بین ویژگی‌های پچ p_i و ویژگی‌های بازسازی پس از ترنسفومر $\hat{p}_{\hat{\sigma}(i)}$ استفاده می‌شود:

$$\mathcal{L}_{rec}\left(p_i, \hat{p}_{\hat{\sigma}(i)}\right) = \left\| \frac{p_i}{\|p_i\|_2} - \frac{\hat{p}_{\hat{\sigma}(i)}}{\|\hat{p}_{\hat{\sigma}(i)}\|_2} \right\|_2^2$$

فرآیندی که در بالا شرح داده شد، برای تشخیص یک پچ تصادفی در تصویر بود. همین فرآیند به سادگی برای چند پچ قابل پیاده‌سازی است. در این صورت برای M پچ تصادفی، ویژگی‌های هرپچ به یک گروه $\frac{N}{M}$ عضوی از جستارهای اشیاء اضافه می‌شود که N تعداد کل جستارها است. برای جلوگیری از ارتباط جستارهای موجود در یک گروه، از ماسک توجه (مشابه DN-DETR) استفاده می‌شود.

۴- ۲- ۲- نتایج

نتایج روش پیش‌آموزش UP-DETR در شکل ۴-۴ نشان می‌دهد که این روش سرعت همگرایی و دقت شبکه DETR را در دیتاست پاسکال افزایش می‌دهد. اگرچه این بهبود در مورد دیتاست کوکو که داده‌های بیشتری دارد، کمتر مشهود است.



(a) AP learning curves of VOC

(b) AP learning curves of COCO

شکل ۴-۴-تاثیر فرآیند پیش‌آموزش UP-DETR در عملکرد شبکه DETR در دو دیتاست کوچک (پاسکال-چپ) و بزرگ(کوکو-راست) [۱۰]

فصل پنجم

جمع‌بندی و نتیجه‌گیری

جمع‌بندی و نتیجه‌گیری

در این گزارش رویکرد جدید تشخیص اشیاء، یعنی استفاده از شبکه ترنسفورمر به همراه تابع هزینه تطبیق دوبخشی(مجارستانی)، معرفی شد. این رویکرد که اولین بار توسط کاریون و همکاران[۹] تحت عنوان شبکهDETR معرفی شد، نیاز به استفاده از اجزا و فرآیندهای از پیش طراحی شده را که پیش از این در همه شبکه‌های تشخیص اشیاء رایج بود، کم‌رنگ کرد. تحت تاثیر از این رویکرد، کارهای بسیار زیادی برای بهبود و حل مشکلات آن معرفی شد. دیدیم که برای حل مشکل همگرایی این شبکه، مقالات متفاوتی سعی در تزریق دانش‌پیشین[۱۲]، بهبود فرآیند آموزش[۱۳] یا تغییر ساختار به منظور کاهش پیچیدگی[۱۱] داشتند. از طرف دیگر، پیچیدگی بالای این شبکه، استفاده از نقشه‌های ویژگی با ابعاد بالا را برای آن غیرممکن می‌کند. برای رفع این مشکل که منجر به ضعف شبکه DETR در تشخیص اشیاء کوچک می‌شود، اولین راه‌حل‌های پیشنهاد شده، کاهش پیچیدگی ماژول توجه در این شبکه بود[۱۱]. با الهام از پیشرفت‌هایی که پیش آموزش بدون نظارت ترنسفورمرها در حوزه پردازش زبان‌های طبیعی ایجاد کرد، پیش‌آموزش بدون نظارت شبکه‌های مبتنی بر DETR برای بهبود آن‌ها امیدوارکننده به نظر می‌رسد[۱۰].

دوسال پس از معرفی DETR و محبوبیت روزافزون آن در بین محققین، درصد بالایی از بهترین مدل‌ها در دیتاست‌هایی مانند کوکو از نسخه‌های بهبود یافته همین شبکه هستند[۱۹]. به نظر می‌رسد سادگی این مدل و قابلیت ذاتی آن برای ترکیب با داده‌های متنی باعث شده این مدل یا ایده‌های آن به سرعت در مسائل دیگر بینایی ماشین مانند قطعه‌بندی[۳۷] و یادگیری محدود[۳۸] و یادگیری چندمودالی[۳۹] [۱۷] رایج شود.

[37] Segmentation

[38] Few-shot learning

[39] Multi-modal

منابع و مراجع

[1] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," *International journal of computer vision*, vol. 128, no. 2, pp. 261-318, 2020.

[2] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the Gap between Anchor-Based and Anchor-Free Detection Via Adaptive Training Sample Selection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9759-9768.

[3] R. Girshick, "Fast R-Cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-Cnn: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in neural information processing systems*, vol. 28, 2015.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

[7] A. Vaswani *et al.*, "Attention Is All You Need," *Advances in neural information processing systems*, vol. 30, 2017.

[8] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171-4186.

[9] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *European conference on computer vision*, 2020: Springer, pp. 213-229.

[10] Z. Dai, B. Cai, Y. Lin, and J. Chen, "Up-Detr: Unsupervised Pre-Training for Object Detection with Transformers," in *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 1601-1610.

[11] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable Detr: Deformable Transformers for End-to-End Object Detection," in *International Conference on Learning Representations*, 2020.

[12] S. Liu *et al.*, "Dab-Detr: Dynamic Anchor Boxes Are Better Queries for Detr," in *International Conference on Learning Representations*, 2021.

[13] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-Detr: Accelerate Detr Training by Introducing Query Denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13619-13627.

[14] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658-666.

[15] T.-Y. Lin *et al.*, "Microsoft Coco: Common Objects in Context," in *European conference on computer vision*, 2014: Springer, pp. 740-755.

[16] J. Dai *et al.*, "Deformable Convolutional Networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764-773.

[17] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1-41, 2022.

[18] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (Voc) Challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303-338, 2010.

[19] " Coco Test-Dev Benchmark." https://paperswithcode.com/sota/object-detection-on-coco (accessed 2023/20/1.