بررسی مدل‌های تشخیص شیء مبتنی بر یادگیری عمیق

محسن نوروزی*، دانشجوی دکتری، حمید حسن پور، استاد، علی قنبری، دانشیار

[۱ و ۲] دانشکده کامپیوتر دانشگاه صنعتی شاهرود شهر شاهرود کشور ایران

[۳] دانشکده کامپیوتر دانشگاه علم و فناوری مازندران شهر بهشهر کشور ایران

چکیده: تشخیص شیء وظیفه طبقه‌بندی و مکان‌یابی اشیاء در یک تصویر یا ویدئو را بر عهده دارد که در سال‌های اخیر به دلیل کاربردهای گسترده آن شهرت یافته است. این مقاله پیشرفت‌های اخیر در بازشناسی شیء مبتنی بر یادگیری عمیق را بررسی می‌کند. مرور کلی مجموعه داده‌های معیار و معیارهای ارزیابی مورد استفاده در شناسایی نیز همراه با برخی از معماری‌های اصلی مورد استفاده در مسئله بازشناسی شیء ارائه شده‌است. همچنین مدل‌های طبقه‌بندی سبک‌وزن مدرن مورد استفاده بررسی شده‌اند. در نهایت، عملکرد این ساختارها را بر روی معیارهای چندگانه مقایسه شده است.

واژه‌های کلیدی: تشخیص شیء و شناسایی، شبکه‌های عصبی پیچشی (CNN)، شبکه‌های سبک‌وزن، یادگیری عمیق

* محسن نوروزی، m.norouzi@shahroodut.ac.ir

A review on object recognition models based on deep learning

Mohsen Norouzi [1,*], Hamid Hassanpour [2] and Ali Ghanbari Sorkhi [3✦*]

[1,2] Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran.

[3] Department of Computer Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran.

Abstract: Object detection is responsible for classifying and locating objects in an image or video, which has become famous in recent years due to its wide applications. This paper reviews recent advances in deep learning-based object recognition. An overview of benchmark datasets and evaluation criteria used in recognition is also presented along with some of the main architectures used in the object recognition problem. Also, the modern lightweight classification models used have been reviewed. Finally, the performance of these structures has been compared on multiple criteria.

Keywords: Object detection and identification, convolutional neural networks (CNN), lightweight networks, deep learning

[*] Corresponding author. Email: m.norouzi@shahroodut.ac.ir

[✦] Email(s): m.norouzi@shahroodut.ac.ir, h.hassanpour@shahroodut.ac.ir, ali.ghanbari@mazust.ac.ir

۱. مقدمه

تشخیص شیء یک کار معمول برای انسان‌ها است. کودک چند ماهه می‌تواند شروع به تشخیص اشیاء کند، با این حال آموزش آن به کامپیوتر تا دهه گذشته کار دشواری بوده‌است. این امر مستلزم شناسایی و بومی‌سازی همه نمونه‌های یک شیء (مانند ماشین‌ها، انسان‌ها، نشانه‌های خیابانی و غیره) در حوزه دید است. به طور مشابه، کارهای دیگر مانند طبقه‌بندی، تقسیم‌بندی، تخمین حرکت، درک صحنه و غیره، مشکلات اساسی در بینایی کامپیوتری بوده‌اند[1].

مدل‌های تشخیص اشیاء اولیه به عنوان مجموعه‌ای از استخراج‌کننده‌های ویژگی دست‌ساز مانند آشکارساز ویولا – جونز[1] [2]، هیستوگرام‌ها (HOG)[2] [3] و غیره ساخته شدند. این مدل‌ها در مجموعه داده‌های ناآشنا به صورت آهسته، نادرست و ضعیف عمل می‌کردند. معرفی شبکه عصبی پیچشی (CNNs)و یادگیری عمیق برای طبقه‌بندی تصویر، چشم‌انداز ادراک بصری را تغییر داد. استفاده از آن در چالش شناخت بصری مقیاس بزرگ نظیر ImageNet (ILSVRC)۲۰۱۲ توسط الکسنت[3] [4] الهام‌بخش تحقیقات بیشتر در مورد کاربرد آن در بینایی ماشین شد. امروزه، تشخیص اشیاء در ماشین‌های خودران و تشخیص هویت در کاربردهای امنیتی و پزشکی کاربرد پیدا می‌کند. در سال‌های اخیر، تشخیص اشیاء با توسعه سریع ابزارها و تکنیک‌های جدید شاهد رشد نمایی بوده‌است. این مقاله مروری یک بررسی جامع از استفاده یادگیری عمیق مبتنی برای بازنمایی اشیاء و معماری طبقه‌بندی سبک انجام داده است. در حالی که بررسی‌های موجود کامل هستند[6] ,[5] ,[1]. بسیاری از آن‌ها فاقد پیشرفت‌های جدید در این حوزه می‌باشند.

در این مقاله، ما به طور سازمان‌یافته ساختارهای مختلف تشخیص شیء و فن‌آوری‌های مرتبط با آن را بررسی کرده‌ایم. در بخش دوم، مساله تشخیص شیء و چالش‌های مرتبط با آن مورد بحث قرار می‌گیرند. مجموعه داده‌های معیار و معیارهای ارزیابی مختلف در بخش سوم فهرست شده‌اند. در بخش چهار، چندین معماری ستون فقرات[4] مورد استفاده در بازنمایی اشیاء مدرن مورد بررسی قرار می‌گیرند. بخش پنجم به سه زیر بخش اصلی تقسیم می‌شود که هر کدام دسته متفاوتی از بازنمایی اشیاء را مورد مطالعه قرار می‌دهند. بخش ششم، دسته خاصی از مدل‌ها بازنمایی اشیاء به نام شبکه‌های سبک‌وزن را بررسی می‌کند. در نهایت بخش مقایسه کارایی و تجزیه و تحلیل در بخش هفتم آورده شده است. نتیجه‌گیری و کارهای آینده در انتها و بخش هشتم بیان می‌شود.

۲. پیشینه

۱،۲. بیان مسئله

بازشناسی شیء گسترش طبیعی طبقه‌بندی شیء است که تنها با هدف تشخیص شیء در تصویر انجام می‌شود. هدف از تشخیص شیء، تشخیص تمام نمونه‌های کلاس‌های از پیش تعریف‌شده و ارائه جایگزیدگی غیردقیق آن در تصویر توسط جعبه‌های هم‌ردیف شده با محور است. آشکار ساز باید قادر به شناسایی تمام نمونه‌های کلاس‌های شیء و رسم جعبه محدوده در اطراف آن باشد. آن به طور کلی به عنوان یک مشکل یادگیری تحت نظارت دیده می‌شود. مدل‌های تشخیص شیء مدرن به مجموعه‌های بزرگی از تصاویر برچسب‌دار برای آموزش دسترسی دارند و بر روی معیارهای متعارف مختلف ارزیابی می‌شوند.

۲،۲. چالش‌های کلیدی در تشخیص شی

بینایی کامپیوتری در دهه گذشته راه درازی را طی کرده‌است، با این حال هنوز هم چالش‌های عمده‌ای برای غلبه بر آن وجود

[1] Viola-Jones

[2] Histogram of Gradients (HOG)

[3] AlexNet

[4] مسیر ارتباطاتی اصلی در یک شبکه گسترده دبیلیوای ان مجموعه کابل‌ها یا اتصالاتی که بیشتر بار ترافیک مخابراتی را حمل می‌کند. سایر مسیرهای داده از این مسیر منشعب می‌شوند

دارد. برخی از این چالش‌های کلیدی که شبکه‌ها در کاربردهای زندگی واقعی با آن‌ها مواجه هستند عبارتند از: تنوع درون طبقه‌ای: تنوع درون طبقه‌ای بین نمونه‌های یک شیء در طبیعت نسبتا رایج است. این تنوع می‌تواند به دلایل مختلفی مانند انسداد، روشنایی، ژست، زاویه دید و غیره باشد. ظاهر خارجی نامحدود می‌تواند اثر چشمگیری بر ظاهر شیء داشته باشد[6]. انتظار می‌رود که اشیاء می‌توانند تغییر شکل غیر سخت داشته باشند یا چرخش داده شوند، مقیاس داده شوند یا تار شوند. برخی اشیاء می‌توانند محیط اطراف نامشخص داشته باشند و استخراج را دشوار سازند. تعداد دسته‌ها: تعداد مطلق کلاس‌های شیء در دسترس برای طبقه‌بندی، حل آن را به یک مساله چالش برانگیز تبدیل می‌کند. همچنین به داده‌های علامت‌گذاری شده با کیفیت بالاتر نیاز دارد، که به سختی می توان از آن استفاده کرد. استفاده از نمونه‌های کمتر برای آموزش آشکارساز یک سوال تحقیقاتی باز است.

۳. مجموعه داده‌ها و معیارهای ارزیابی

۱,۳. مجموعه داده

این بخش یک نمای کلی از مجموعه داده‌هایی که در دسترس هستند را ارائه می‌دهد و به طور معمول برای وظایف تشخیص اشیاء استفاده می‌شود.

۱) PASCAL VOC07/12: چالش کلاس‌های شیء بصری پاسکال (VOC)یک تلاش چند ساله برای شتاب بخشیدن به توسعه در زمینه ادراک بصری بود. این کار در سال ۲۰۰۵ با وظایف طبقه‌بندی و تشخیص بر روی چهار کلاس شیء آغاز شد [7]، اما دو نسخه از این چالش‌ها را که عمدتا به عنوان یک معیار استاندارد در نظر گرفته می‌شوند مورد استفاده قرار دادیم. در حالی که چالش VOC07 دارای پنج هزار تصویر آموزشی و بیش از ۱۲ هزار شیء با برچسب [8] بود، چالش VOC12 آن‌ها را به ۱۱ هزار تصویر آموزشی و بیش از ۲۷ هزار شیء با برچسب [9] افزایش داد. کلاس‌های شیء به ۲۰ دسته گسترش داده شدند و وظایفی مانند تقسیم‌بندی و تشخیص عمل نیز در نظر گرفته شدند. پاسکال VOC میانگین میانگین دقت[1] (mAP) را در ۰,۵ IoU[2] (سطح مقطع بر روی اجتماع) برای ارزیابی عملکرد مدل‌ها معرفی کرد.

۲) ILSVRC: چالش شناخت بصری با مقیاس بزرگ ImageNet (ILSVRC)[10] یک چالش سالانه در حال اجرا از سال ۲۰۱۰ تا ۲۰۱۷ بود و به معیاری برای ارزیابی عملکرد الگوریتم تبدیل شد. اندازه مجموعه داده تا بیش از یک میلیون تصویر متشکل از ۱۰۰۰ کلاس طبقه‌بندی شیء مقیاس‌بندی شد. منابع مختلفی از جمله ImageNet [11] و Flikr، برای ساخت مجموعه داده شناسایی مورد استفاده قرار گرفتند. ILSVRC همچنین معیار ارزیابی را با کاهش آستانه IoU برای کمک به شناسایی اشیای کوچکتر به معرفی می‌کند.

۳) MS - COCO: اشیاء مشترک مایکروسافت در زمینه (MS - COCO) [12] یکی از چالش برانگیزترین مجموعه داده موجود است. ۹۱ شیء مشترک در بافت طبیعی آن‌ها یافت می‌شود که یک انسان ۴ ساله به راحتی می‌تواند آن‌ها را تشخیص دهد. این مجموعه در سال ۲۰۱۵ راه‌اندازی شد و محبوبیت آن به مرور افزایش‌یافته است. مجموعه داده بیش از دو میلیون نمونه و به طور متوسط ۳,۵ دسته در هر تصویر دارد. علاوه بر این، این روش شامل متوسط ۷,۷ شیء در هر تصویر است، که به راحتی بیشتر از سایر مجموعه داده‌های محبوب است. COCO MS شامل تصاویری از زوایای مختلف نیز می‌باشد. این مجموعه داده یک روش دقیق‌تر برای اندازه‌گیری عملکرد آشکارساز معرفی کرد. بر خلاف پاسکال VOC و ILSVCR، IoT را از ۰,۵ تا ۰,۹۵ در گام‌های ۰,۰۵ محاسبه می‌کند، سپس از ترکیبی از این ۱۰ مقدار به عنوان متریک نهایی استفاده می‌کند که به آن دقت متوسط (AP) می‌گویند. علاوه بر این، AP برای اشیاء کوچک، متوسط و بزرگ به طور جداگانه محاسبه می‌شود که برای

[1] Mean Average Precession (mAP)

[2] Intersection over Union (IoU)

مقایسه عملکرد در مقیاس‌های مختلف قابل استفاده است [5, 13].

۲,۳. معیارها

بازنمایی اشیاء از معیارهای چندگانه برای اندازه‌گیری عملکرد آشکارسازها استفاده می‌کنند. با این حال، میانگین دقت (mAP)رایج‌ترین معیار ارزیابی است. دقت از اشتراک بر اجتماع (IoU)به دست می‌آید، که نسبت مساحت همپوشانی و مساحت اجماع داده مرجع و کادر مرزی پیش‌بینی‌شده می‌باشد. آستانه‌ای تنظیم می‌شود تا مشخص شود که آیا تشخیص صحیح است یا خیر. اگر IoU بیشتر از آستانه باشد، به عنوان مثبت واقعی طبقه‌بندی می‌شود در حالی که یک IoU در زیر آستانه به عنوان مثبت کاذب طبقه‌بندی می‌شود. اگر مدل نتواند یک شیء موجود در داده مرجع را تشخیص دهد، آن را منفی کاذب می‌نامند. دقت، درصد پیش‌بینی‌های صحیح را اندازه‌گیری می‌کند در حالی که یادآوری، پیش‌بینی‌های صحیح را با توجه به داده مرجع اندازه‌گیری می‌کند. روابط (۱) و (۲) نحوه محاسبه دقت و فراخوانی را نشان می‌دهد.

$$\text{دقت} = \frac{\text{مثبت واقعی}}{\text{مثبت کاذب} + \text{مثبت واقعی}} = \frac{\text{مثبت واقعی}}{\text{همه مشاهدات}} \quad (۱)$$

$$\text{فراخوانی} = \frac{\text{مثبت واقعی}}{\text{منفی کاذب} + \text{مثبت واقعی}} = \frac{\text{مثبت واقعی}}{\text{تمام داده مرجع}} \quad (۲)$$

براساس معادله بالا، دقت متوسط برای هر کلاس به طور جداگانه محاسبه می‌شود. برای مقایسه عملکرد بین آشکارسازها، میانگین دقت میانگین همه کلاس‌ها، به نام میانگین دقت (mAP) استفاده می‌شود، که به عنوان یک معیار برای ارزیابی نهایی عمل می‌کند[5].

۴. معماری‌های ستون فقرات[1]

معماری‌های ستون فقرات یکی از مهم‌ترین اجزای آشکارساز شیء هستند. این شبکه‌ها ویژگی تصویر ورودی مورد استفاده توسط مدل را استخراج می‌کنند. در اینجا ما برخی از معماری‌های ستون فقرات مورد استفاده در آشکارسازهای مدرن را مورد بحث قرار داده‌ایم:

AlexNet .۱,۴

AlexNet از هشت لایه قابل تعلیم، پنج لایه پیچشی و سه لایه کاملا متصل تشکیل شده‌است. آخرین لایه از لایه کاملا متصل به یک طبقه‌بندی کننده N - way (N: تعداد کلاس‌ها) سافت‌ماکس (بیشینه هموار) متصل شده‌است. این روش از هسته‌های پیچشی چندگانه در سراسر شبکه برای به دست آوردن ویژگی‌های تصویر استفاده می‌کند. همچنین از افت فشار و RELU[2] به ترتیب برای تنظیم و هم‌گرایی آموزش سریع‌تر استفاده می‌کند [14]. در حالی که AlexNet و جانشینان مشابه اش [15] بر اندازه پنجره دریافتی کوچک‌تر برای بهبود دقت تمرکز کرده‌اند، سیمونیان و زیسرمن اثرات عمق شبکه بر روی آن را بررسی کرده‌اند. آن‌ها VGG[3] [16] را پیشنهاد کردند که از فیلترهای پیچشی کوچک برای ساخت شبکه‌های با عمق‌های مختلف استفاده می‌کرد. در حالی که یک میدان دریافتی بزرگ‌تر را می‌توان با مجموعه‌ای از فیلترهای پیچشی کوچک‌تر به دست آورد، پارامترهای شبکه را به شدت کاهش می‌دهد و زودتر همگرا می‌شود. این مقاله نشان داد که چگونه معماری شبکه عمیق (۱۶ – ۱۹ لایه) می‌تواند برای انجام طبقه‌بندی و مکان‌یابی با دقت بالا مورد استفاده قرار گیرد.

ResNets.۲,۴

همانطور که شبکه‌های عصبی پیچشی عمیق‌تر و عمیق‌تر می‌شوند، نگارندگان در [17] نشان دادند که چگونه دقت آن‌ها ابتدا اشباع می‌شود و سپس به سرعت کاهش می‌یابد. آن‌ها

[1] Back-Bone architectures

[2] Rectified Linear Unit

[3] Visual Geometry Group

استفاده از یادگیری باقی مانده برای لایه‌های انباشت شده را برای کاهش افت عملکرد پیشنهاد کردند. این امر با اضافه کردن یک اتصال فرار بین لایه‌ها محقق می‌شود. این اتصال یک اضافه عنصری بین ورودی و خروجی بلوک است و پارامتر یا پیچیدگی محاسباتی اضافی به شبکه اضافه نمی‌کند. شبکه ResNet با ۳۴ لایه معمولی [17] اساسا یک فیلتر پیچشی بزرگ (۷×۷) و به دنبال آن ۱۶ ماژول گسترش (جفت فیلترهای ۳×۳ کوچک با میان‌بر شناسایی بر روی آن‌ها) و در نهایت یک لایه کاملا متصل است. ساختار تنگناها را می‌توان با انباشت لایه‌های پیچشی ۳ (۳×۱، ۳×۳، ۱×۱) به جای ۲، برای شبکه‌های عمیق‌تر تطبیق داد.

نگارندگان این پژوهش همچنین نشان دادند که چگونه شبکه ۱۶ لایه VGG پیچیدگی بالاتری نسبت به معماری ResNet لایه‌های ۱۰۱ و ۱۵۲ عمیق‌تر خود دارد در حالی که دقت کمتری دارد. ResNetها به طور گسترده در ستون فقرات طبقه بندی و شناسایی استفاده می‌شوند و اصول اصلی آن الهام بخش بسیاری از شبکه‌ها بوده است.

GoogLeNet/Inception.۳,۴

سگدی و همکاران اتلاف محاسبات در شبکه را به عنوان یک دلیل اصلی برای آن فرض کردند. مدل‌های بزرگ‌تر همچنین تعداد زیادی پارامتر دارند و تمایل به بیش از حد متناسب با داده‌ها دارند. آن‌ها استفاده از معماری متصل پراکنده محلی را به جای معماری کاملا متصل برای حل این مسائل پیشنهاد کردند. بنابراین GoogLeNet یک شبکه عمیق ۲۲ لایه‌ای است که با توده کردن ماژول‌های دریافتی متعدد در بالای یکدیگر ایجاد شده‌است. ماژول‌های دریافتی شبکه‌هایی هستند که چندین فیلتر با اندازه یکسان دارند. نقشه‌های ویژگی ورودی از این فیلترها عبور کرده و الحاق شده و به لایه بعدی ارسال می‌شوند. این شبکه همچنین دارای دسته‌بندی کننده‌های کمکی در لایه‌های میانی برای کمک به منظم کردن و انتشار گرادیان می‌باشد[18]. GoogLeNet نشان داد که چگونه استفاده کارآمد از بلوک‌های محاسباتی می‌تواند همانند دیگر شبکه‌های پارامتر سنگین عمل کند. این روش بدون داده‌های خارجی به دقت ۹۳,۳ ٪ top-5 در مجموعه داده ImageNet [10] دست یافت، در حالی که از سایر مدل‌های معاصر سریع‌تر بود. نسخه‌های به روز شده Inception مانند [20] ,[19] نیز در سال‌های بعد منتشر شد که عملکرد آن را بیشتر بهبود بخشید و شواهد بیشتری از کاربردهای معماری‌های به هم پیوسته اصلاح شده ارائه داد.

ResNeXt.۴,۴

روش‌های متداول موجود برای بهبود دقت یک مدل یا با افزایش عمق و یا عرض مدل بود. با این حال، افزایش هر یک از این موارد منجر به پیچیدگی بیشتر مدل و تعداد پارامترها می‌شود، در حالی که حاشیه سود به سرعت کاهش می‌یابد. زی و همکاران معماری ResNeXt [21] را معرفی کردند که ساده‌تر و کارآمدتر از دیگر مدل‌های موجود است. ResNeXt از انباشته شدن بلوک‌های مشابه در VGG/ResNet [17]و رفتار «تقسیم تبدیل–ادغام» ماژول Inception [20] الهام گرفت. این اساسا یک ResNet است که در آن هر بلوک ResNet با یک ماژول شروع مانند ResNeXt جایگزین می‌شود. ماژول‌های تبدیل پیچیده و مناسب از دریافتی با ماژول‌های مشابه در بلوک‌های ResNeXt جایگزین می‌شوند، که مقیاس بندی و تعمیم شبکه را آسان‌تر می‌کند. زی و همکاران همچنین تاکید می‌کنند که کاردینالیتی مسیرهای توپولوژیکی در بلوک ResNeXt می‌تواند به عنوان یک بعد سوم، همراه با عمق و عرض، برای بهبود دقت مدل در نظر گرفته شود.

CSPNett.۵,۴

شبکه‌های عصبی موجود نتایج قابل توجهی را در دستیابی به دقت بالا در وظایف بینایی ماشین نشان داده‌اند. با این حال،

آن‌ها به منابع محاسباتی بیش از حد متکی هستند. وانگ[1] و همکاران بر این باورند که محاسبات استنباطی سنگین را می‌توان با کاهش اطلاعات گرادیان تکراری در شبکه کاهش داد. آن‌ها CSPNet[2] [22] را پیشنهاد کردند که مسیرهای مختلفی را برای جریان گرادیان درون شبکه ایجاد می‌کند. CSPNet نقشه‌های ویژگی را در لایه پایه به دو بخش تقسیم می‌کند. یک بخش از بلوک شبکه پیچشی جزئی (به عنوان مثال، بلوک متراکم و انتقال در DenseNet[3] [23] یا بلوک Res(X) در ResNeXt [21]) عبور داده می‌شود، در حالی که بخش دیگر با خروجی‌های آن در مرحله بعد ترکیب می‌شود. این امر تعداد پارامترها را کاهش می‌دهد، استفاده از واحدهای محاسباتی را افزایش می‌دهد و مدیریت حافظه را آسان‌تر می‌کند. اجرای آن آسان و به طور کلی و به اندازه کافی برای قابل‌اجرا بودن در معماری‌های دیگر مانند ResNet ، ResNeXt ،DenseNet ، scaled − YOLOv4 [24] و غیره قابل پیاده‌سازی است. استفاده از CSPNet در این شبکه‌ها محاسبات را از ۱۰ ٪ به ۲۰ ٪ کاهش داد، در حالی که دقت ثابت باقی ماند یا بهبود یافت. هزینه حافظه و تنگنای محاسباتی نیز با این روش به طور قابل‌توجهی کاهش می‌یابد. در نتیجه از آن در بسیاری از مدل‌های آشکارساز پیشرفته استفاده می‌شود، در حالی که برای دستگاه‌های موبایل و لبه[4] نیز استفاده می‌شود.

EfficientNet.۶,۴

تان و همکاران به طور سازمان‌یافته مقیاس شبکه و اثرات آن بر عملکرد مدل را مطالعه کردند. آن‌ها به طور خلاصه بیان کردند که چگونه تغییر پارامترهای شبکه مانند عمق، عرض و وضوح بر دقت آن تاثیر می‌گذارد. مقیاس گذاری هر پارامتر به صورت جداگانه با هزینه مرتبط همراه است. افزایش عمق یک شبکه می‌تواند به جذب ویژگی‌های غنی‌تر و پیچیده‌تر کمک کند، اما آموزش آن‌ها به دلیل مشکل گرادیان محو شونده دشوار است. به طور مشابه، مقیاس گذاری عرض شبکه ثبت ویژگی‌های ریز مقیاس را آسان‌تر می‌کند اما در به دست آوردن ویژگی‌های سطح بالا مشکل دارد. از افزایش وضوح تصویر، مانند عمق و عرض، به عنوان مقیاس‌های مدل اشباع می‌شود[25]. در [25] استفاده از یک ضریب ترکیبی را پیشنهاد شد که می تواند هر سه بعد را به طور یکنواخت مقیاس کند. هر پارامتر مدل یک ثابت مرتبط دارد، که با ثابت نگه داشتن ضریب به صورت ۱ و انجام جستجوی شبکه‌ای در یک شبکه پایه یافت می‌شود. معماری پایه، با الهام از کار قبلی آن‌ها [26]، با جستجوی معماری عصبی در یک هدف جستجو و در عین حال بهینه‌سازی دقت و محاسبات توسعه داده شده‌است.

جدول (۱): مقایسه معماری‌های ستون فقرات

مدل	سال	لایه‌ها	تعداد پارامترها (ملیون)	دقت به درصد
AlexNet	۲۰۱۲	۷	۶۲,۴	۶۳,۳
VGG-16	۲۰۱۴	۱۶	۱۳۸,۴	۷۳
GoogLeNet	۲۰۱۴	۲۲	۶,۷	-
ResNet-50	۲۰۱۵	۵۰	۲۵,۶	۷۶
ResNeXt-50	۲۰۱۶	۵۰	۲۵	۷۷,۸
CSPRResNeXt-50	۲۰۱۹	۵۹	۲۰,۵	۷۸,۲
EfficientNet-B4	۲۰۱۹	۱۶۰	۱۹	۸۳

۵. بازنمای اشیاء

ما این بررسی را براساس دو نوع از آشکارسازها – آشکارسازهای دو مرحله‌ای و تک مرحله‌ای – تقسیم کرده‌ایم. با این حال، ما همچنین روش‌های پیشگام را مورد بحث قرار دادیم، که در آن به طور خلاصه به بررسی برخی از بازنمایی اشیاء سنتی می‌پردازیم. شبکه‌ای که یک ماژول جداگانه برای تولید پیشنهادها منطقه‌ای دارد به عنوان یک آشکارساز دو مرحله‌ای شناخته می‌شود. این مدل‌ها تلاش می‌کنند تا تعداد دلخواهی از طرح‌های پیشنهادی اشیاء در یک تصویر را در

[1] Wang

[2] Cross Stage Partial Network

[3] Densely Connected Convolutional Networks

[4] Edge devices

مرحله اول پیدا کنند و سپس آن‌ها را در مرحله دوم طبقه‌بندی و مکان‌یابی کنند.

۱،۵. روش‌های پیشگام

۱) «ویولا جونز[1]» در ابتدا برای تشخیص چهره طراحی شده‌بود، آشکارساز شیء ویولا جونز ، که در سال ۲۰۰۱ ارائه شد، یک آشکارساز دقیق و قدرتمند بود. این روش چندین تکنیک مانند ویژگی‌های مانند هاار[2]، تصویر کامل، طبقه‌بندی کننده آدابوست[3] و آبشاری را ترکیب می‌کند. مرحله اول جستجوی ویژگی‌ها، شبیه به روش هاار با لغزش یک پنجره بر روی تصویر ورودی و استفاده از تصویر انتگرالی انجام می‌شود. سپس از یک آدابوست آموزش دیده برای یافتن طبقه‌بندی کننده ویژگی هاار و آبشاری آن‌ها استفاده می‌کند[2].

۲) آشکارساز HOG[4]. در سال ۲۰۰۵، هیستوگرام شعاع‌های جهتی (HOG) توسط [27] پیشنهاد شد که برای استخراج ویژگی‌ها برای تشخیص شیء مورد استفاده قرار می‌گیرند. این یک بهبود نسبت به آشکارسازهای دیگر مانند [31]–[28] بود. گرادیان HOG و جهت گیری آن برای ایجاد یک جدول ویژگی را استخراج می‌کند. تصویر به شبکه‌ها تقسیم می‌شود و سپس جدول ویژگی برای ایجاد هیستوگرام برای هر سلول در شبکه مورد استفاده قرار می‌گیرد. ویژگی‌های HOG برای ناحیه موردنظر تولید شده و برای تشخیص به یک طبقه‌بندی کننده ماشین بردار پشتیبان (SVM)[5] خطی تغذیه می‌شوند. این آشکارساز برای ردیابی عابران پیاده پیشنهاد شده‌است. با این حال، می‌توان کلاس‌های مختلف را به آن آموزش داد.

۳) DPM[6]. مدل قطعات تغییر شکل پذیر در [32] معرفی شد و برنده چالش پاسکال VOC[7] در سال ۲۰۰۹ بود. این روش از «بخشی» منفرد از شیء برای تشخیص استفاده کرد و دقت بالاتری نسبت به HOG به دست آورد. فلسفه تقسیم و غلبه را دنبال می‌کند؛ بخشی از شیء به صورت جداگانه در زمان استنباط تشخیص داده می‌شود و آرایش احتمالی آن به صورت تشخیص مشخص می‌شود. مدل‌های مبتنی بر DPM [34] ,[33] یکی از موفق‌ترین الگوریتم‌ها قبل از دوره یادگیری عمیق بودند.

۲،۵. آشکارسازهای دومرحله ای

۱) R-CNN[8]. شبکه عصبی پیچشی مبتنی بر منطقه اولین مقاله در خانواده R-CNN بود[35] و نشان داد که چگونه می‌توان از CNNها برای بهبود عملکرد تشخیص استفاده کرد. R- CNN از ماژول پیشنهادی منطقه کلاس آگنوستیک با CNN برای تبدیل تشخیص به مشکل طبقه بندی و محلی سازی استفاده می‌کند. تصویر ورودی متوسط کاهیده ابتدا از ماژول پیشنهاد ناحیه عبور می‌کند، که ۲۰۰۰ نامزد هدف را تولید می‌کند. این ماژول بخش‌هایی از تصویر را پیدا می‌کند که احتمال بیشتری برای پیدا کردن یک شیء با استفاده از جستجوی انتخابی دارد[36]. سپس این نامزدها از طریق شبکه CNN تکثیر می‌شوند که یک بردار ویژگی ۴۰۹۶ بعدی را برای هر پیشنهاد استخراج می‌کند.

R- CNN یک فرآیند آموزشی پیچیده چند مرحله‌ای دارد. مرحله اول پیش آموزش CNN با مجموعه داده طبقه‌بندی بزرگ است. سپس با جایگزین کردن لایه طبقه‌بندی با یک طبقه‌بندی کننده *N+1*-way که به طور تصادفی مقدار دهی شده، N تعداد کلاس‌ها با استفاده از کاهش گرادیان تصادفی (SGD)[9] [37]

[1] Viola Jones

[2] Haar

[3] Adaboost

[4] Histogram of Gradients (HOG)

[5] Support Vector Machine

[6] Deformable Part Model

[7] Visual Object Classes

[8] Regions with Convolutional Neural Network

[9] Stochastic gradient descent

برای تشخیص با استفاده از تصاویر منحصر به فرد (پیشنهادهای میانگین کاهیده، منحرف)به خوبی تنظیم می‌شود. یک SVM خطی و رگرسیون جعبه مرزی برای هر کلاس آموزش داده شده‌است.

۲) SPP-Net. هی[1] و همکارانش استفاده از لایه ادغام هرم فضایی (SPP) برای پردازش تصویر با اندازه دلخواه یا نسبت ابعاد دلخواه را پیشنهاد کردند. آن‌ها متوجه شدند که تنها بخش کاملا متصل CNN به یک ورودی ثابت نیاز دارد. SPP - net [38] صرفا لایه‌های پیچشی CNN را قبل از ماژول پیشنهاد منطقه تغییر داد و یک لایه ادغام اضافه کرد، در نتیجه شبکه را از نسبت اندازه / ابعاد مستقل کرد و محاسبات را کاهش داد. الگوریتم جستجوی انتخابی [36] برای تولید پنجره‌های کاندید استفاده می‌شود. نقشه‌های ویژگی با عبور از تصویر ورودی از طریق لایه‌های پیچشی یک شبکه ZF -5 [15] به دست می‌آیند. پنجره‌های کاندید سپس بر روی نقشه‌های ویژگی ترسیم می‌شوند، که متعاقبا توسط فایل‌های اجرایی فضایی یک لایه ادغام هرمی به بازنمایی‌های طول ثابت تبدیل می‌شوند. این بردار به لایه کاملا متصل و در نهایت به طبقه بنده‌ای SVM برای پیش‌بینی کلاس و امتیاز منتقل می‌شود. همانند R - CNN [35]، SPP - net[2] به عنوان لایه پس پردازش برای بهبود محلی‌سازی با رگرسیون جعبه محدود است. همچنین از همان فرآیند آموزش چندمرحله‌ای استفاده می‌کند، با این تفاوت که تنظیم دقیق تنها بر روی لایه‌های کاملا متصل انجام می‌شود.

۳)R-CNN سریع: یکی از مشکلات اصلی R-CNN/SPPNet نیاز به آموزش چندین سیستم به طور جداگانه بود. Fast R-CNN [39] این مشکل را با ایجاد یک سیستم منفرد آموزش پذیر نقطه به نقطه حل کرد. شبکه به عنوان ورودی یک تصویر و طرح پیشنهادی شیء آن در نظر گرفته می‌شود. تصویر از طریق مجموعه‌ای از لایه‌های پیچشی عبور داده می‌شود و طرح‌های پیشنهادی شیء به نقشه‌های ویژگی به‌دست‌آمده نگاشت می‌شوند. ژیشیک ساختار هرمی لایه‌های ادغام را از SPP - net [38] با یک لایه فضایی به نام لایه ادغام RoI[3] جایگزین کرد. این لایه به ۲ لایه کاملا متصل شده متصل شده و سپس به یک لایه سافت‌ماکس کلاس *N+1* و یک لایه رگرسور بسته، که دارای یک لایه کاملا متصل است، تقسیم می‌شود. این مدل همچنین تابع اتلاف بازگردی جعبه محدود کننده از L2 را برای هموارسازی L1 به عملکرد بهتر تغییر داد، در حالی که یک اتلاف چند وظیفه‌ای را برای آموزش شبکه معرفی کرد. نویسندگان از نسخه اصلاح‌شده مدل‌های موجود از پیش آموزش‌دیده مانند [40] به عنوان ستون فقرات استفاده کردند.

R-CNN سریع به عنوان یک بهبود در سرعت (۱۴۶ برابر در R-CNN) در حالی که افزایش دقت مکمل آن بود. این روش تمرین ساده شده، ادغام هرمی را حذف کرده و یک تابع زیان جدید معرفی می‌کند. آشکار ساز شیء، بدون شبکه پیشنهاد منطقه، با دقت قابل‌توجهی در زمان واقعی نزدیک به سرعت گزارش شده‌است.

۴) R-CNN سریعتر: حتی با وجود اینکه R-CNN سریع به تشخیص شیء در زمان واقعی نزدیکتر بود، تولید پیشنهاد منطقه آن همچنان یک مرتبه کندتر بود (۲ ثانیه در هر تصویر در مقایسه با ۰,۲ ثانیه در هر تصویر). رن[4] و همکاران یک شبکه کاملا پیچیده [41] را به عنوان یک شبکه پیشنهاد ناحیه (RPN[5]) پیشنهاد کردند که یک تصویر ورودی دل‌خواه را می‌گیرد و مجموعه‌ای از پنجره‌های کاندید را خروجی می‌دهد. هر یک از این پنجره‌ها دارای یک نمره عینی مرتبط هستند که احتمال یک شیء را تعیین می‌کند. RPN برخلاف پیشینیان خود مانند [39] که از هرم‌های تصویر برای حل واریانس اندازه اشیاء استفاده می‌کنند، جعبه مهار یا لنگر[6] را معرفی می‌کند. این روش از جعبه‌های متعدد محدود کننده از نسبت‌های ابعاد مختلف استفاده

[1] He

[2] spatial pyramid pooling to remove the fixed-size constraint of the network

[3] Region of Interest

[4] Ren

[5] Region Proposal Network

[6] Anchor

کرده و برای مکان‌یابی شیء بر روی آن‌ها رگرسیون داده شده‌است. تصویر ورودی ابتدا از طریق CNN عبور داده می‌شود تا مجموعه‌ای از نقشه‌های ویژگی به دست آید. این موارد به RPN که جعبه‌های مرزی و طبقه‌بندی آن‌ها را تولید می‌کند، ارسال می‌شوند. سپس پیشنهادهای منتخب به نقشه‌های ویژگی به‌دست‌آمده از لایه CNN قبلی در لایه ادغام RoI، و در نهایت به لایه کاملا متصل، که به طبقه بند و رگرسیون جعبه محدود فرستاده می‌شود، ارسال می‌شوند. R-CNN سریعتر اساساً Fast R-CNN با RPN به عنوان ماژول پیشنهاد منطقه است.

آموزش سریعتر R-CNN به دلیل وجود لایه‌های مشترک بین دو مدل که وظایف بسیار متفاوتی را انجام می‌دهند پیچیده‌تر است. در مرحله اول، RPN در مجموعه داده ImageNet از قبل آموزش‌دیده [11] و در مجموعه داده PASCAL VOC به خوبی تنظیم شده‌است[7]. یک R-CNN سریع از اولین مرحله از پیشنهادات منطقه‌ای RPN آموزش دیده است. تا این نقطه، شبکه‌ها لایه پیچشی مشترک ندارند. دراین مرحله، تلفیق لایه‌های آشکارساز معرفی می‌شود که در نتیجه آن لایه‌های منحصر به فرد RPN به خوبی تنظیم می‌گردد.

۵)FPN استفاده از هرم تصویر برای به دست آوردن هرم ویژگی (یا هرم‌های تصویر خاص) در سطوح مختلف روشی رایج برای افزایش تشخیص اشیاء کوچک است. حتی اگر دقت میانگین آشکار ساز را افزایش دهد، افزایش در زمان استنتاج قابل‌توجه است. لین و همکارانش، شبکه هرم ویژه (FPN) [42] را پیشنهاد نموده‌اند که دارای یک معماری از بالا به پایین با اتصالات جانبی برای ساخت ویژگی‌های معنایی سطح بالا در مقیاس‌های مختلف می‌باشد. FPN دو مسیر دارد، یک مسیر از پایین به بالا که یک سلسله مراتب ویژگی محاسباتی ConvNet در چندین مقیاس است، و یک مسیر از بالا به پایین که نقشه‌های مشخصه غیردقیق را از سطح بالاتر به ویژگی‌های با وضوح بالا نمونه‌برداری می‌کند. این مسیرها با اتصال جانبی توسط یک عملیات پیچشی 1×1 به هم متصل می‌شوند تا اطلاعات معنایی در ویژگی‌ها افزایش یابد. FPN می‌تواند معنای سطح بالایی را در همه مقیاس‌ها فراهم کند که نرخ خطا در تشخیص را کاهش می‌دهد. این امر به یک بلوک ساختمان استاندارد در مدل‌های تشخیص آینده تبدیل شد و دقت آن‌ها را در سراسر جدول بهبود بخشید. همچنین منجر به توسعه شبکه‌های بهبود یافته دیگری مانند PANet [43]، NAS - FPN [44] و EfficientNet [45] می‌شود که آشکارساز پیشرفته فعلی است.

۶) R-FCN[1]. دای[2] و همکاران، شبکه پیچشی کامل مبتنی بر منطقه (R - FCN) [46] را پیشنهاد کردند که تقریبا تمام محاسبات درون شبکه را، بر خلاف آشکارسازهای دو مرحله‌ای قبلی که تکنیک‌های متمرکز بر منابع را در هر پیشنهاد به کار می‌بردند، به اشتراک گذاشت. آن‌ها در مقابل استفاده از لایه‌های کاملا متصل بحث کردند و در عوض از لایه‌های پیچشی استفاده کردند. با این حال، لایه‌های عمیق‌تر در شبکه پیچشی، انتقال ناپذیر هستند و آنها را برای وظایف بومی‌سازی بی‌اثر می‌سازد. آشکارساز R- FCN ترکیبی از چهار شبکه پیچشی است. تصویر ورودی ابتدا از ResNet -101 [17] عبور داده می‌شود تا نقشه‌های ویژگی به دست آید. خروجی میانی (لایه) به شبکه پیشنهاد منطقه (RPN) منتقل می‌شود تا پیشنهادها ROI را شناسایی کند در حالی که خروجی نهایی بیشتر از طریق یک لایه پیچشی پردازش می‌شود و ورودی طبقه‌بندی کننده و بازگردنده است. لایه طبقه‌بندی نقشه حساس به موقعیت تولید شده را با پیشنهادات ROI ترکیب می‌کند تا پیش بینی‌ها را ایجاد کند در حالی که شبکه رگرسیون جزئیات جعبه مرزی را خروجی می‌دهد.

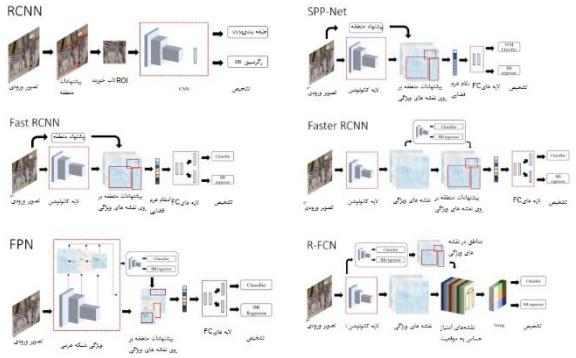
۷)Mask R - CNN Mask R - CNN [47] با اضافه کردن شاخه دیگری به صورت موازی برای بخش‌بندی نمونه شیء در سطح پیکسل، بر روی R-CNN سریع‌تر گسترش می‌یابد. این شاخه یک شبکه کاملا متصل است که بر روی ROIs اعمال می‌شود تا هر پیکسل را به بخش‌هایی با هزینه محاسباتی کلی کم طبقه‌بندی کند. این روش از معماری اولیه مشابه R-CNN سریع‌تر برای پیشنهاد شیء استفاده می‌کند، اما یک سر ماسک را

[1] Regional Fully connected Network

[2] Dai

به موازات طبقه‌بندی و سر رگرسور جعبه محدود اضافه می‌کند. یک تفاوت عمده استفاده از لایه RoIAlign به جای لایه RoIPol برای جلوگیری از انحراف سطح پیکسل ناشی از کوانتیزه سازی فضایی بود. نویسندگان، ResNeXt - 101 [48] را به عنوان ستون فقرات خود همراه با شبکه هرمی مشخصه (FPN) برای دقت و سرعت بهتر انتخاب کردند. تابع اتلاف Faster R - CNN با اتلاف ماسک به روز رسانی می‌شود و همانند FPN از پنج جعبه مهار با نسبت ابعاد سه استفاده می‌کند. آموزش کلی Mask R-CNN شبیه به R-CNN سریع‌تر است.

۸)DetectoRS : بسیاری از آشکارسازهای دو مرحله‌ای جدیدتر مانند [50] ,[49] ,[41] از مکانیسم نگاه کردن و تفکر دو بار استفاده می‌کنند یعنی محاسبه پیشنهادها شیء اول و استفاده از آن‌ها برای استخراج ویژگی‌ها برای تشخیص اشیاء. مدل [51] این مکانیزم را هم در سطح ماکرو و هم در سطح میکرو شبکه به کار می‌گیرد. در سطح کلان، آن‌ها هرم ویژگی بازگشتی[1] (RFP) را پیشنهاد می‌کنند که از طریق توده کردن شبکه هرم ویژگی چندگانه[2] (FPN) با اتصال بازخورد اضافی از مسیر سطح بالا به پایین در FPN به لایه پایین به بالا تشکیل شده‌است. خروجی FPN قبل از عبور از آن به لایه بعدی FPN، توسط لایه Poling هرم فضایی Atrous (ASPP)[3] [52] پردازش می‌شود. یک ماژول فیوژن برای ترکیب خروجی‌های FPN از ماژول‌های مختلف با ایجاد یک نقشه توجه مورد استفاده قرار می‌گیرد. در سطح میکرو، کیائو[4] و همکارانش، تبدیل Atrous قابل تعویض (SAC)[5]را برای تنظیم نرخ انبساط پیچشی ارائه کردند. یک لایه ادغام متوسط با فیلتر ۵×۵ و یک پیچش ۱×۱ به عنوان یک تابع سوئیچ برای تصمیم‌گیری در مورد نرخ پیچش استفاده می‌شود [53]، که به تشخیص ستون فقرات اشیاء در مقیاس مختلف در پرواز کمک می‌کند. آن‌ها همچنین SAC را بین دو ماژول بافت عمومی بسته‌بندی کردند [54] زیرا به ایجاد سوئیچینگ پایدارتر کمک می‌کند. ترکیب این دو تکنیک، هرم فرکانس بازگشتی و تبدیل فرکانس قابل تعویض، منجر به آشکارسازی می‌شود. شکل (۱) نحوه کار آشکارسازهای دو مرحله‌ای را در چند معماری مختلف نشان می‌دهد.



شکل (۱): تصویر معماری داخلی اشیاء دو مرحله‌ای مختلف تشخیص شی

۳،۵. جداساگرهای تک مرحله ای

۱)آشکارسازهای تک مرحله‌ای YOLO[6] تشخیص شیء را به عنوان یک مساله طبقه‌بندی حل می‌کنند، یک ماژول برخی از کاندیدهایی را ارائه می‌دهد که شبکه آن‌ها را به عنوان یک شیء یا پس‌زمینه طبقه‌بندی می‌کند. با این حال، YOLO یا «شما فقط نگاه کنید» [55] به طور مستقیم پیکسل‌های تصویر را به عنوان اشیاء و ویژگی‌های جعبه محدود کننده آن پیش‌بینی می‌کند. در YOLO، تصویر ورودی به یک شبکه S×S تقسیم می‌شود و سلولی که مرکز شیء در آن قرار می‌گیرد مسئول تشخیص آن است. یک سلول شبکه‌ای، چندین باکس‌های مرزی را پیش‌بینی می‌کند و هر آرایه پیش‌بینی شامل ۵ عنصر است: مرکز جعبه – x و y، ابعاد جعبه – w و h و امتیاز اطمینان.

YOLO از مدل GoogLeNet برای طبقه‌بندی تصویر الهام گرفت، که از ماژول‌های آبشاری شبکه‌های پیچشی کوچک‌تر استفاده می‌کند [19]. این روش بر روی داده‌های ImageNet از پیش آموزش داده شده‌است تا زمانی که مدل به دقت بالایی

[1] Recursive feature pyramid

[2] Feature Pyramid Network

[3] Atrous Spatial Pyramid Pooling

[4] Qiao

[5] Soft Actor Critic

[6] You Only Look Once

دست یافته و سپس با اضافه کردن پیچش اولیه تصادفی و لایه‌های کاملا متصل، اصلاح گردد. در زمان آموزش، سلول‌های شبکه تنها یک کلاس را با هم‌گرایی بهتر پیش‌بینی می‌کنند، اما در طول زمان استنتاج افزایش می‌یابد. از دست دادن مولتیچ، از دست دادن ترکیبی تمام مولفه‌های پیش‌بینی‌شده، برای بهینه‌سازی مدل استفاده می‌شود. سرکوب غیر حداکثر (NMS) تشخیص‌های چندگانه مختص کلاس را حذف می‌کند. YOLO مدل‌های زمان واقعی تک مرحله‌ای معاصر خود را با یک حاشیه بزرگ در هر دو دقت و سرعت پشت سر گذاشت. با این حال، کاستی‌های قابل‌توجهی نیز داشت. دقت محلی‌سازی برای اشیاء کوچک یا خوشه‌ای و محدودیت تعداد اشیاء در هر سلول از اشکالات عمده آن بود. این مسائل در نسخه‌های بعدی YOLO [59]–[56] ثابت شدند.

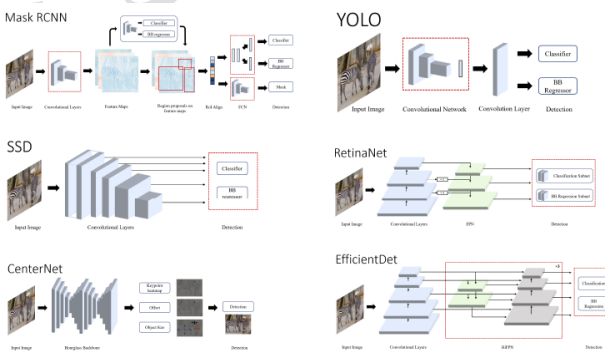
۲)آشکارساز مولتی باکس تک شات SSD[1] [60] اولین آشکارساز تک مرحله‌ای بود که با دقت آشکارسازهای دو مرحله‌ای مانند Faster R - CNN مطابقت داشت، در حالی که سرعت زمان واقعی را حفظ می‌کرد. SSD بر روی VGG - 16 با ساختارهای کمکی اضافی برای بهبود عملکرد ساخته شد. این لایه‌های پیچشی کمکی، که به انتهای مدل اضافه شده‌اند، به تدریج اندازه خود را کاهش می‌دهند. SSD اشیاء کوچک‌تر را پیش از این در شبکه شناسایی می‌کند، زمانی که ویژگی‌های تصویر خیلی خام نیستند، در حالی که لایه‌های عمیق‌تر مسئول جبران جعبه‌های پیش‌فرض و نسبت‌های ابعاد می‌باشند[61].

در طول آموزش، SSD هر جعبه حقیقی پایه مولتی باکس با جعبه‌های پیش‌فرض با بهترین همپوشانی جاکارد مطابقت می‌دهد و شبکه را مطابق با آن آموزش می‌دهد، مشابه با multibox [61] آن‌ها همچنین از استخراج منفی سخت و افزایش داده‌های سنگین استفاده کردند. مشابه با DPM[2]، از مجموع وزنی جایگزیدگی و اتلاف اطمینان برای آموزش مدل استفاده کرد. خروجی نهایی با اجرای سرکوب غیر بیشینه به دست می‌آید.

۳)YOLOv2 و YOLO9000: YOLOv2، یک موازنه آسان بین سرعت و دقت ارائه می‌دهد در حالی که مدل YOLO9000 می‌تواند ۹۰۰۰ کلاس شیء را در زمان واقعی پیش‌بینی کند. آن‌ها معماری ستون فقرات GoogLeNet را با darknet-19 جایگزین کردند. این روش بسیاری از تکنیک‌های تاثیرگذار مانند نرمال سازی دسته‌ای [62] را برای بهبود هم‌گرایی، آموزش مشترک سیستم‌های طبقه‌بندی و تشخیص به منظور افزایش کلاس‌های تشخیص، حذف لایه‌های کاملا متصل به منظور افزایش سرعت و استفاده از جعبه‌های لنگر یادگرفته برای بهبود فراخوانی و داشتن برش‌های بهتر ترکیب کرده‌است.

YOLOv2 انعطاف‌پذیری بهتری را برای انتخاب مدل سرعت و دقت فراهم کرد و معماری جدید پارامترهای کمتری داشت. همانطور که عنوان مقاله نشان می‌دهد، «بهتر، سریع‌تر و قوی‌تر» بود [56].

۴)RetinaNet با توجه به تفاوت بین دقت آشکارسازهای تک و دو مرحله‌ای، لین و همکاران پیشنهاد کردند که دلیل تاخیر آشکارسازهای تک مرحله‌ای «عدم تعادل کلاس پیش‌زمینه شدید» است[63]. آن‌ها یک تابع هزینه آنتروپی متقاطع تغییر شکل یافته را پیشنهاد کردند، که در آن از دست دادن کانونی را به عنوان ابزاری برای رفع عدم تعادل می‌نامند. پارامتر اتلاف متمرکز، سهم اتلاف را از مثال‌های ساده کاهش می‌دهد. نویسندگان کارایی آن را با کمک یک آشکارساز ساده و تک مرحله‌ای به نام RetinaNet نشان می‌دهند [63]، که اشیاء را با نمونه‌برداری متراکم تصویر ورودی در محل، مقیاس و نسبت ابعاد پیش‌بینی می‌کند.



شکل (۲): تصویر معماری داخلی اشیاء دو مرحله‌ای مختلف تشخیص شیء

[1] single-shot detector

[2] Deformable Parts Model

۵) YOLOv3: YOLOv3 دارای "بهبود افزایشی" از نسخه‌های قبلی YOLO بود [56] ,[55]. ردم و همکاران، شبکه استخراج ویژگی را با یک شبکه دارکر – ۵۳ بزرگ‌تر جایگزین کردند. آن‌ها همچنین تکنیک‌های مختلفی مانند افزایش داده، آموزش چندمقیاسی، نرمال سازی دسته‌ای را در میان دیگر روش‌ها گنجانده‌اند. در لایه طبقه‌بندی کننده با یک طبقه‌بندی کننده لجستیک، softmax جایگزین شد[57].

۶) CenterNet: پژوهشگران در [5] به جای نمایش جعبه محدود مرسوم، رویکرد بسیار متفاوتی از اشیاء مدل‌سازی به عنوان نقطه اتخاذ کرده‌اند. CenterNet شیء را به عنوان یک نقطه واحد در مرکز جعبه مرزی پیش‌بینی می‌کند. تصویر ورودی از طریق FCN عبور می‌کند که یک نقشه گرما تولید می‌کند که قله‌های آن با مرکز شیء کشف‌شده متناظر است. این روش از یک Hourglass - 101 پشته شده ImageNet از پیش آموزش‌دیده به عنوان شبکه استخراج ویژگی استفاده کرده و دارای ۳ سر نقشه جهت تعیین مرکز شیء، سر بعد جهت تخمین اندازه شیء و سر افست جهت اصلاح افست نقطه شیء می‌باشد[64].

۷) EfficientDet : این مدل به سمت ایده آشکارساز مقیاس پذیر با دقت و بازده بالاتر ایجاد می‌شود. این مدل ویژگی‌های چند مقیاسی کارآمد، BiFPN[1] و مقیاس بندی مدل را معرفی می‌کند. BiFPN یک شبکه هرمی دو جهته با وزن‌های قابل یادگیری برای اتصال متقابل ویژگی‌های ورودی در مقیاس‌های مختلف می‌باشد. این روش در NAS - FPN بهبود می‌یابد، که به آموزش سنگین نیاز دارد و شبکه پیچیده‌ای دارد، با حذف نوده‌ای تک ورودی و اضافه کردن یک اتصال جانبی اضافی. این امر گره‌های کم‌تر کارآمد را حذف کرده و ادغام ویژگی سطح بالا را افزایش می‌دهد. برخلاف آشکارسازهای موجود که با لایه‌های FPN بزرگ‌تر، عمیق‌تر یا توده بندی شده مقیاس گذاری می‌کنند، افعانکت یک ضریب ترکیب معرفی می‌کند که می‌تواند برای "مقیاس بندی مشترک تمام ابعاد شبکه ستون فقرات، شبکه BiFPN، شبکه کلاس / جعبه و وضوح" مورد استفاده قرار گیرد[45]. تلاش Det با استفاده از EFicientNet به عنوان شبکه ستون فقرات با چندین مجموعه از لایه‌های BiFPN که به صورت سری به عنوان شبکه استخراج ویژگی قرار گرفته‌اند، عمل می‌کند. هر خروجی از لایه BiFPN نهایی به شبکه پیش‌بینی کلاس و جعبه ارسال می‌شود. این مدل با استفاده از بهینه‌ساز SGD[2] همراه با نرمال‌سازی دسته‌ای همگام شده آموزش داده می‌شود و به جای فعال‌سازی استاندارد ReLU که متمایز، کارآمدتر است و عملکرد بهتری دارد، از فعال‌سازی swish استفاده می‌کند [65].

۸): YOLOv4ایده‌های جالب زیادی را برای طراحی یک آشکارساز شیء سریع و آسان برای آموزش که می‌تواند در سیستم‌های تولید موجود کار کند، ترکیب کرد. این روش از "کیف وسایل رایگان" استفاده می‌کند، یعنی روش‌هایی که تنها زمان آموزش را افزایش می‌دهند و بر زمان استنباط تاثیر نمی‌گذارند. YOLOv4 [58] از تکنیک‌های افزایش داده، روش‌های تنظیم، هموارسازی برچسب کلاس، CIoT U - loss ، نرمال‌سازی (CmBN)[3]، آموزش خود – خصمانه، نزول گرادیان تصادفی با شروع مجدد گرم[66] و سایر ترفندها برای بهبود آموزش استفاده می‌کند. روش‌هایی که تنها بر زمان استنباط تاثیر می‌گذارند، به نام "کیسه نمونه‌های"، به شبکه اضافه می‌شوند، از جمله فعال‌سازی مش [67]، اتصالات جزئی چند مرحله‌ای (CSP) ، بلوک SPP - SPP، بلوک تجمعی مسیر PAN[4]، ارتباطات باقیمانده چند ورودی وزن دار و غیره. همچنین از الگوریتم ژنتیک برای جستجوی بیش از حد استفاده کردند.

این روش دارای ستون فقرات CSPNetdarknet - 53 از پیش آموزش‌دیده ImageNet، گردن بلوک SPP و PAN و سر تشخیص YOLOv2 می‌باشد.

[1] Bi directional FPN

[2] Stochastic Gradient Descent

[3] Cross Mini Batch Network

[4] Path Aggregation Network

۹) Swin Transformer: مبدل‌ها [68] تاثیر عمیقی در حوزه پردازش زبان طبیعی (NLP) از زمان آغاز آن داشته‌اند. کاربرد آن در مدل‌های زبانی مانند berT (نمایندگی رمزگذار دو جهته از مبدل‌ها)[69] ، GPT (ترانسفورمر پیش‌آموزش عمومی) [70]، T5 (ترانسفورمر انتقال متن به متن) [71]و غیره باعث پیشرفت هنر در این زمینه شده‌است. مبدل‌ها [68] از مدل توجه برای ایجاد وابستگی‌های میان عناصر توالی استفاده می‌کنند و می‌توانند نسبت به سایر معماری‌های متوالی، به بافت طولانی‌تر توجه کنند. موفقیت ترانسفورماتورها در NLP[1] باعث ایجاد علاقه به استفاده از آن در بینایی ماشین شد. در حالی که CNNها ستون فقرات پیشرفت در دید بوده‌اند، آن‌ها برخی کاستی‌های ذاتی مانند عدم اهمیت زمینه جهانی، وزن ثابت پس از آموزش [21] و غیره را دارند.

تبدیل Swin [72] به دنبال فراهم کردن یک ستون فقرات مبتنی بر ترانسفورماتور برای وظایف بینایی ماشین است. این روش تصاویر ورودی را در تکه‌های متعدد و غیر هم پوشان تقسیم می‌کند و آن‌ها را به تعبیه تبدیل می‌کند. سپس بلوک‌های تبدیل Swin متعددی در ۴ مرحله به قطعات اعمال می‌شوند، که هر مرحله متوالی تعداد قطعات را برای حفظ نمایش سلسله مراتبی کاهش می‌دهد. بلوک تبدیل Swin از ماژول‌های خود – توجه چند سر محلی ، براساس پنجره وصله جابجا شده متناوب در بلوک‌های متوالی تشکیل شده‌است. پیچیدگی محاسبات با اندازه تصویر در خود – توجه محلی خطی می‌شود در حالی که پنجره جابجا شده اتصال میان پنجره را ممکن می‌سازد. همچنین نشان می‌دهد که چگونه پنجره‌های جابه‌جا شده دقت تشخیص را با سربار کم افزایش می‌دهند. شکل (۲) معماری دسته تک مرحله‌ای را در مقایسه با دو مرحله‌ای نشان می‌دهد. براساس توضیحات در شکل(۲) چند نمونه از این معماری‌های مشهور نشان داده شده است.

۶.شبکه‌های سبک

یک شاخه جدید از تحقیقات در سال‌های اخیر با هدف طراحی شبکه‌های کوچک و کارآمد برای محیط‌های با منابع محدود شکل‌گرفته است، همانطور که در به‌کارگیری اینترنت اشیاء (IoT) رایج است[75]–[73]. این روند به طراحی بازنمایی اشیاء قوی نیز نفوذ کرده‌است. مشاهده می‌شود که اگر چه تعداد زیادی از بازنمایی اشیاء به دقت عالی دست می‌یابند و استنتاج را در زمان واقعی انجام می‌دهند، اکثر این مدل‌ها به منابع محاسباتی بیش از حد نیاز دارند و بنابراین نمی‌توانند بر روی دستگاه‌های لبه مستقر شوند.

بسیاری از رویکردهای مختلف نتایج هیجان انگیزی را در گذشته نشان داده‌اند. استفاده از اجزای کارآمد و تکنیک‌های فشرده‌سازی مانند هرس [78]–[76]، کوانتیزه سازی[79] ,[77] و غیره کارایی مدل‌های یادگیری عمیق را بهبود بخشیده‌است. استفاده از شبکه بزرگ آموزش‌دیده برای آموزش مدل‌های کوچک‌تر، به نام تقطیر [80]، نیز نتایج جالبی را نشان داده‌است.

۱،۶. SqueezeNet

پیشرفت‌های اخیر در زمینه CNNها عمدتا بر بهبود دقت سطح بالای مجموعه داده‌های معیار متمرکز بود که منجر به انفجار اندازه مدل و پارامترهای آن‌ها شد. اما در سال ۲۰۱۶، ایاندولا[2] و همکاران یک شبکه کوچک‌تر و هوشمندتر به نام SqeezeNet را پیشنهاد کردند که ضمن حفظ عملکرد، پارامترها را کاهش داد. آن‌ها با استفاده از سه استراتژی طراحی اصلی به این هدف دست یافتند. با استفاده از فیلترهای کوچک‌تر، تعداد کانال‌های ورودی به ۳ × ۳ فیلتر کاهش می‌یابد و پس از آن لایه‌های پایین گذر در شبکه قرار می‌گیرند. دو استراتژی اول تعداد پارامترها را هنگام تلاش برای حفظ دقت کاهش می‌دهند و استراتژی سوم دقت شبکه را افزایش می‌دهد. بلوک ساختمان SqeezeNet یک ماژول آتش نامیده می‌شود که از دو لایه تشکیل شده‌است: یک لایه فشاری و یک لایه بسط یافته، هر کدام با یک فعال‌سازی ReLU. لایه squeeze از فیلترهای چندگانه یک در یک ساخته شده‌است در حالی که لایه گسترش، ترکیبی از فیلترهای یک در

[1] Natural Language Processing

[2] Iandola

یک و سه در سه است، در نتیجه تعداد کانال‌های ورودی محدود می‌شود. معماری SqeezeNet از مجموعه‌ای از ۸ ماژول آتش تشکیل‌شده که بین لایه‌های پیچشی قرار گرفته‌اند. با الهام از ResNet، SqeezeNet با اتصالات باقی مانده نیز پیشنهاد شد که دقت را بر روی مدل افزایش داد[5].

MobileNets .۲,۶

MobileNet [73] از روش‌های مرسوم مدل‌های کوچک مانند کوچک شدن، هرس کردن، کوانتیزه کردن یا فشرده‌سازی دور شد و در عوض از معماری شبکه کارآمد استفاده کرد. این شبکه یک پیچش استاندارد از کرنل‌ها بر روی تمام کانال‌های ورودی استفاده می‌کند و آن‌ها را در یک مرحله با هم ترکیب می‌کند، در حالی که پیچش چند جمله‌ای از کرنل‌های مختلف برای هر کانال ورودی استفاده می‌کند و از پیچش نقطه به نقطه برای ترکیب ورودی‌ها استفاده می‌کند. این جداسازی فیلترینگ و ترکیب ویژگی‌ها، هزینه محاسبات و اندازه مدل را کاهش می‌دهد. MobileNet شامل ۲۸ لایه پیچشی مجزا است که هر کدام به دنبال نرمال‌سازی دسته‌ای و تابع فعالسازی ReLU قرار دارند. هاوارد[1] و همکارانش همچنین دو پارامتر کوچک شدگی مدل را معرفی کرده‌اند: ضرب‌کننده عرض و قدرت تفکیک به منظور بهبود بیشتر سرعت و کاهش اندازه مدل مورد استفاده قرار گرفت. ضرب‌کننده عرض، عرض شبکه را به طور یکنواخت با کاهش کانال‌های ورودی و خروجی دستکاری می‌کند در حالی که ضرب‌کننده رزولوشن، اندازه تصویر ورودی و بازنمایی‌های آن را در سراسر شبکه تحت‌تاثیر قرار می‌دهد. مبولیت نت به دقت قابل قیاس با برخی مدل‌های تکامل‌یافته دست می‌یابد در حالی که کسری از اندازه آن‌ها است. هاوارد و همکاران همچنین نشان دادند که چگونه آن می‌تواند در کاربردهای مختلف مانند اسناد چهره، مفهوم‌سازی جغرافیایی و تشخیص شیء تعمیم داده شود. با این حال، بسیار ساده و خطی مانند VGG بود و بنابراین راه‌های کمتری برای جریان گرادیان داشت. این موارد در تکرارهای بعدی این مدل [75] ,[74] تثبیت شدند.

ShuffleNet .۳,۶

ژانگ[2] و همکارانش در سال ۲۰۱۷، ShuffleNet [76] را معرفی کردند که یک معماری شبکه عصبی بسیار کارآمد از نظر محاسباتی است که به طور خاص برای دستگاه‌های تلفن همراه طراحی شده‌است. آن‌ها متوجه شدند که بسیاری از شبکه‌های کارآمد با کاهش مقیاس و وارد کردن آن به دلیل پیچیدگی ۱×۱ پر هزینه، کم‌تر موثر می‌شوند. در ارتباط با کانال، آن‌ها استفاده از پیچش گروهی را برای دور زدن اشکال آن در جریان محدود اطلاعات پیشنهاد کردند. ShuffleNet عمدتا از یک پیچش استاندارد و به دنبال آن پشته‌ای از واحدهای ShuffleNet که در سه مرحله گروه‌بندی شده‌اند، تشکیل شده‌است. واحد ShuffleNet شبیه به بلوک ResNet است که در آن از پیچش عمقی در لایه ۳×۳ استفاده می‌کنند و لایه یک در یک را با پیچش گروه نقطه به نقطه جایگزین می‌کنند. لایه پیچشی عمقی با یک عملیات شافل کانالی حمایت که پیش از آن قرار دارد، حمایت می‌شود. هزینه محاسبه ShuffleNet را می‌توان با دو ابرپارامتر مدیریت کرد: عدد گروهی برای کنترل پراکندگی اتصال و فاکتور مقیاس بندی برای دستکاری اندازه مدل. با بزرگ شدن تعداد گروه‌ها، نرخ خطا با کاهش کانال‌های ورودی به هر گروه، اشباع می‌شود و بنابراین ممکن است قابلیت‌های نمایشی را کاهش دهد. ShuffleNet نسبت به مدل‌های معاصر عملکرد بهتری داشت، در حالی که اندازه آن به طور قابل‌توجهی کوچک‌تر بود. از آنجا که تنها پیشرفت در ShuffleNet، کشیده شدن کانال بود، هیچ پیشرفتی در سرعت استنتاج مدل وجود ندارد.

MobileNetv2 .۴,۶

[1] Howard

[2] Zhang

بهبود در MobileNetv1 [73]، سندلر[1] و همکاران در سال ۲۰۱۸، MobileNetv2 [74] را پیشنهاد دادند. این روش، باقیمانده معکوس را با گلکوگاه خطی، یک ماژول لایه جدید برای کاهش محاسبات و بهبود دقت، معرفی می‌کند. این ماژول، برخلاف بلوک باقی مانده معمولی که عملیات فشرده‌سازی، پیچش و سپس انبساط را انجام می‌دهد، یک نمایش با ابعاد پایین از ورودی به ابعاد بالا را گسترش می‌دهد. MobileNetv2 شامل یک لایه پیچشی و به دنبال آن ۱۹ ماژول گلوگاه باقی مانده و در نتیجه دو لایه پیچشی است. ماژول گلوگاه باقیمانده تنها زمانی دارای یک اتصال میان‌بر است که گام یک باشد. برای گام‌های بیشتر، به دلیل تفاوت در ابعاد، از میان‌بر استفاده نمی‌شود. آن‌ها همچنین از REL به عنوان تابع غیر خطی، به جای ReLU ساده، برای محدود کردن محاسبات استفاده کردند. برای تشخیص شیء، نویسندگان از MobileNetv2 به عنوان استخراج‌کننده ویژگی متغیر از نظر محاسباتی کارآمد SSD استفاده کردند [60].

PeleeNet .۵,۶

مدل‌های یادگیری عمیق سبک موجود مانند خانواده MobileNet به شدت به پیچش جدا شدنی وابسته بودند که در نتیجه فاقد پیاده‌سازی کارآمد می‌شدند. وانگ[2] و همکارانش با استفاده از مجموعه‌ای از تکنیک‌های حفظ محاسبات، یک معماری کارآمد جدید را براساس پیچش مرسوم به نام PeleeNet [78] ارائه نموده‌اند. PeleeNet در اطراف DenseNet متمرکز بود اما به بسیاری از مدل‌های دیگر برای الهام نگاه کرد. این روش لایه‌های متراکم دو طرفه، بلوک ساقه، تعداد دینامیک کانال‌ها در یک تنگنای، تراکم لایه انتقال و پس فعال‌سازی معمولی را برای کاهش هزینه محاسبات و افزایش سرعت معرفی می‌کند. لایه متراکم دو طرفه به دریافت مقیاس‌های مختلف میدان دریافتی کمک می‌کند و شناسایی اشیاء بزرگ‌تر را آسان‌تر می‌کند. برای کاهش از دست دادن اطلاعات، از یک بلوک ساقه به همان روش [79] استفاده شد.

ShuffleNetv2 .۶,۶

در سال ۲۰۱۸، ما[3] و همکاران مجموعه‌ای از دستورالعمل‌های جامع را برای طراحی معماری‌های شبکه کارآمد در ShuffleNetv2 ارائه دادند [77]. آن‌ها برای استفاده از معیارهای مستقیم مانند سرعت یا تاخیر برای اندازه‌گیری پیچیدگی محاسباتی، به جای معیارهای غیر مستقیم مانند گل، استدلال کردند. ShuffleNetv2 براساس چهار اصل راهنما ساخته شده‌است: ۱) عرض برابر برای کانال‌های ورودی و خروجی به منظور به حداقل رساندن هزینه دسترسی به حافظه، ۲) انتخاب دقیق پیچش گروهی براساس پلتفرم و وظیفه هدف، ۳) ساختارهای چندمسیره به دقت بالاتری در هزینه بهره‌وری دست می‌یابند و ۴) عملیات آلمان مانند اضافه کردن و ReLU از نظر محاسباتی قابل‌چشم‌پوشی نیستند. با پیروی از اصول فوق، آن‌ها یک بلوک ساختمانی جدید طراحی کردند. این روش ورودی را با استفاده از یک لایه تقسیم کانال به دو بخش تقسیم می‌کند و به دنبال آن سه لایه پیچش قرار دارند که سپس با اتصال باقی مانده ادغام می‌شوند و از طریق یک لایه لگام کانال عبور می‌کنند. برای مدل نمونه‌برداری نزولی، تقسیم کانال حذف می‌شود و اتصال باقی مانده دارای لایه‌های پیچشی تفکیک‌پذیر است.

MnasNet .۷,۶

با افزایش نیاز به مدل‌های دقیق، سریع و با تاخیر کم برای دستگاه‌های لبه مختلف، طراحی چنین شبکه عصبی چالش برانگیزتر از همیشه شده‌است. در سال ۲۰۱۸، تان[4] و همکاران MnasNet [26] را پیشنهاد کردند که از یک رویکرد جستجوی معماری عصبی خودکار (NAS) طراحی شده‌بود. آن‌ها مساله جستجو را به عنوان بهینه‌سازی چند منظوره با هدف دقت بالا و

[1] Sandler

[2] Wang

[3] Ma

[4] Tan

تاخیر کم فرمول‌بندی کردند. همچنین فضای جستجو را با پارتیشن بندی CNN به بلوک‌های منحصر به فرد و سپس جستجو برای عملیات و ارتباطات در آن بلوک‌ها به طور جداگانه فاکتورگیری می‌کند و در نتیجه فضای جستجو را کاهش می‌دهد. این امر همچنین به هر بلوک این امکان را می‌دهد که طراحی متمایزی داشته باشد، برخلاف مدل‌های قبلی که همان بلوک‌ها را روی هم انباشته کرده‌اند. نویسندگان از عامل یادگیری تقویتی مبتنی بر RNN به عنوان کنترل‌کننده به همراه یک مربی برای اندازه‌گیری دقت و دستگاه‌های سیار برای تاخیر استفاده کردند. هر مدل نمونه‌برداری شده بر روی یک کار آموزش داده می‌شود تا دقت خود را به دست آورده و بر روی دستگاه‌های واقعی برای تاخیر اجرا شود. این امر برای رسیدن به یک هدف پاداش نرم استفاده می‌شود و کنترلر به روز می‌شود. این فرآیند تا زمانی تکرار می‌شود که حداکثر تکرارها یا یک نامزد مناسب به دست آید.

MobileNetv3 ۸,۶

در قلب MobileNetv3 [75] همان روش مورد استفاده برای ایجاد MnasNet [26] با برخی اصلاحات است. یک پلتفرم آگاه از جستجوی ساختار عصبی خودکار در یک فضای جستجوی سلسله مراتبی فاکتورگیری شده اجرا می‌شود و در نتیجه توسط NetAdapt [81] بهینه می‌شود، که مولفه‌های مورد استفاده شبکه را در تکرارهای متعدد حذف می‌کند. هنگامی که یک طرح پیشنهادی معماری به دست می‌آید، کانال‌ها را سه‌گانه می‌کند، به طور تصادفی وزن‌ها را آغاز می‌کند و سپس آن را تنظیم می‌کند تا معیارهای هدف را بهبود بخشد. این مدل بعدا برای حذف برخی از لایه‌های گران‌قیمت در معماری و به دست آوردن بهبود تاخیر اضافی اصلاح شد. هاوارد[1] و همکاران استدلال کردند که فیلترهای معماری اغلب تصاویر بازتابی از یکدیگر هستند، و این دقت را می‌توان حتی پس از حذف نیمی از این فیلترها حفظ کرد. استفاده از این تکنیک محاسبات را کاهش داد. MobileNetv3 از ترکیبی از ReLU و Hard Swish به عنوان توابع فعال‌سازی استفاده کرد، دومی عمدتا در انتهای مدل به کار گرفته می‌شود. Hard Swish تفاوت قابل‌توجهی با تابع Swish معمولی ندارد اما از نظر محاسباتی سبک‌تر است در حالی که دقت حفظ می‌شود. برای موارد استفاده از منابع مختلف , [75] دو مدل MobileNetv3 - Large و MobileNetv3 - Small را معرفی کرد. MobileNetv3 - Large از ۱۵ بلوک گلوگاه تشکیل شده‌است در حالی که MobileNetv3 - Small ۱۱ واحد از این نوع بلوک را دارد. همچنین این شبکه‌ها دارای لایه فشار و تحریک [54] در بلوک‌های ساختمانیشان هستند. مشابه با [74]، این مدل به عنوان یک آشکارساز ویژگی در SSDLite عمل می‌کند و ۳۵ ٪ سریع‌تر از مدل‌های قبلی است در حالی که به mAP بالاتری دست می‌یابد.

جدول (۲): مقایسه عملکرد آشکارسازهای شیای مختلف بر روی مجموعه داده‌های MS COCO و PASCAL VOC ۲۰۱۲ در اندازه تصویر ورودی مشابه.

مدل	سال	ستون فقرات	اندازه	AP[0.5:0.95]	AP0.5	FPS
R-CNN*	2014	AlexNet	224	-	58.50%	~0.02
SPP-Net*	2015	ZF-5	Variable	-	59.20%	~0.23
Fast R-CNN*	2015	VGG-16	Variable	-	65.70%	~0.43
Faster R-CNN*	2016	VGG-16	600	-	67.00%	5
R-FCN	2016	ResNet-101	600	31.50%	53.20%	~35
FPN	2017	ResNet-101	800	36.20%	59.10%	5
Mask R-CNN	2018	ResNeXt-101-FPN	800	39.80%	62.30%	~4
DetectoRS	2020	ResNeXt-101	1333	53.30%	71.60%	
YOLO*	2015	(Modified) GoogLeNet	448	-	57.90%	45
SSD	2016	VGG-16	300	23.20%	41.20%	46
YOLOv2	2016	DarkNet-19	352	21.60%	44.00%	81
RetinaNet	2018	ResNet-101-FPN	400	31.90%	49.50%	12
YOLOv3	2018	DarkNet-53	320	28.20%	51.50%	45
CenterNet	2019	Hourglass-104	512	42.10%	61.10%	7.8
EfficientDet-D2	2020	Efficient-B2	768	43.00%	62.30%	41.7
YOLOv	20	CSPDark	512	43.00%	64.9	31

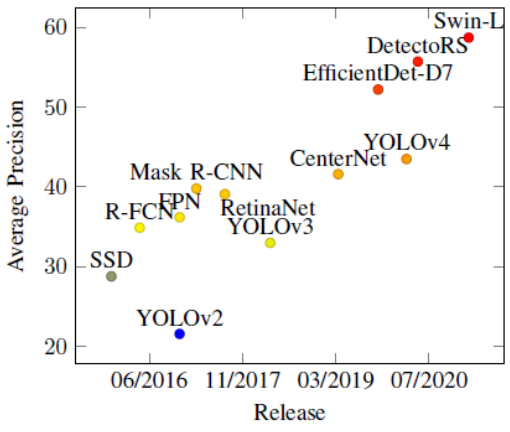
[1] Howard

4	20	Net-53			0%	
Swin-L	20 21	HTC++	-	57.70%	-	-

۷. نتایج مقایسه

ما عملکرد هر دو آشکارساز تک و دو مرحله‌ای را در مجموعه داده‌های PASCAL VOC ۲۰۱۲ و مایکروسافت COCO مقایسه می‌کنیم. عملکرد بازنمایی اشیاء تحت‌تاثیر عواملی همچون اندازه و مقیاس تصویر ورودی، استخراج کننده ویژگی، معماری GPU، تعداد طرح‌های پیشنهادی، روش آموزش، تابع اتلاف و غیره می‌باشد که مقایسه مدل‌های مختلف بدون محیط معیار مشترک را دشوار می‌سازد. در اینجا در جدول (۲)، عملکرد مدل‌ها را براساس نتایج مقالات آن‌ها ارزیابی می‌کنیم. مدل‌ها بر روی دقت متوسط (AP)و فریم‌های پردازش‌شده در هر ثانیه (FPS)در زمان استنتاج مقایسه می‌شوند.

AP. مجموعه داده COCO یک معیار عملکرد دیگر AP [۰,۵ :۰,۹۵]، یا به سادگی AP، که میانگین AP برای IoU از ۰,۵ تا ۰,۹۵ در اندازه گام ۰,۵ است را معرفی کرد. ما عمدا عملکرد آشکارسازها را بر روی تصویر ورودی با اندازه مشابه، در صورت امکان، برای ارائه یک حساب منطقی مقایسه می‌کنیم، همانطور که نویسندگان اغلب یک آرایه از مدل‌ها را برای ارائه انعطاف‌پذیری بین دقت و زمان استنتاج معرفی می‌کنند. در شکل (۳)، ما تنها از مدل مدرن از آرایه احتمالی خانواده آشکارسازی شیء مدل‌ها استفاده می‌کنیم.



شکل (۳): عملکرد بازنمایی اشیاء در مجموعه داده MS COCO.

۸. نتیجه گیری

حتی اگر تشخیص شیء در دهه گذشته راه طولانی را طی کرده باشد، بهترین آشکارسازها هنوز هم از اشباع در عملکرد دور هستند. با افزایش کاربردهای آن در دنیای واقعی، نیاز به مدل‌های سبک‌وزن که بتوانند بر روی سیستم‌های موبایل و تعبیه‌شده گسترش یابند، به صورت نمایی افزایش می‌یابد. علاقه زیادی به این حوزه وجود داشته‌است، اما هنوز هم یک چالش آشکار است. در این مقاله، ما نشان داده‌ایم که چگونه آشکارسازهای دو مرحله‌ای و تک مرحله‌ای نسبت به آشکارسازهای قبلی خود توسعه یافته‌اند. در حالی که آشکارسازهای دو مرحله‌ای به طور کلی دقیق‌تر هستند، کند هستند و نمی توان از آن‌ها برای کاربردهای بلادرنگ مانند اتومبیل شخصی یا امنیت استفاده کرد. با این حال، این موضوع در چند سال اخیر تغییر کرده‌است که در آن یک آشکارساز یک مرحله‌ای به همان اندازه دقیق و بسیار سریع‌تر از قبلی است. همانطور که در شکل ۳ مشهود است، مبدل Swin دقیق‌ترین آشکارساز تا به امروز است اما مدل‌های مبتنی بر YOLO اگرچه دقت کمتری دارند اما با ابتکارات در معماری و دارا بودن نرخ بررسی فریم بهتر جایگاهی را به خود اختصاص داده‌اند که ادامه توسعه آن‌ها امیدبخش می‌کند. از سوی دیگر رویکردهای مبتنی بر ستون فقرات سنگین همچنان بهترین پاسخ‌ها را بر مبنای دقت به همراه دارند. با روند مثبت فعلی در دقت آشکارسازها، ما امید زیادی به آشکارسازهای دقیق‌تر و سریع‌تر داریم.

منابع

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, 2023. *https://doi.org/10.1109/JPROC.2023.3238524*.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. neural Inf. Process. Syst.*, pp. 1097–1105, 2012.

[15] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, 2021, pp. 8748–8763.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Comput. Sci.*, vol. 2, no. 3, p. 160, 2021. *https://doi.org/10.1007/s42979-021-00592-x*.

[19] L. Alzubaidi *et al.*, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. big Data*, vol. 8, pp. 1–74, 2021. *https://doi.org/10.1186/s40537-021-00444-8*.

[20] F. Chen, J. Wei, B. Xue, and M. Zhang, "Feature fusion and kernel selective in Inception-v4 network," *Appl. Soft Comput.*, vol. 119, p. 108582, 2022.*https://doi.org/10.1016/j.asoc.2022.108582*.

[21] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–41, 2022. *https://doi.org/10.1145/3505244*.

[22] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[23] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3523–3542, 2021. *https://doi.org/10.1109/TPAMI.2021.3059968*.

[24] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," in *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 2021, pp. 13029–13038.

[25] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.

[26] M. Tan *et al.*, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, 2001, vol. 1, pp. I–I.

[3] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. big data*, vol. 6, no. 1, pp. 1–48, 2019. *https://doi.org/ 10.1186/s40537-019-0197-0*.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017. *https://doi.org/ 10.1145/3065386*.

[5] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digit. Signal Process.*, vol. 126, p. 103514, 2022. *https://doi.org/ 10.1016/j.dsp.2022.103514*.

[6] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, pp. 261–318, 2020. *https://doi.org/ 10.1007/s11263-019-01247-4*.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010. *https://doi.org/ 10.1007/s11263-009-0275-4*.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results http://www. pascal-network. org/challenges," in *VOC/voc2007/workshop/index. html*, 2007.

[9] M. Everingham and J. Winn, "The PASCAL visual object classes challenge 2012 (VOC2012) development kit," *Pattern Anal. Stat. Model. Comput. Learn. Tech. Rep*, vol. 2007, no. 1–45, p. 5, 2012.

[10] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015. *https://doi.org/ 10.1007/s11263-015-0816-y*

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255. *https://doi.org/10.1109/ CVPR.2009.5206848*

[12] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision--ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014, pp. 740–755.

[13] A. Kuznetsova *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1956–1981, 2020. *https://doi.org/10.1007/s11263-020-01316-z*.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015. *https://doi.org/10.1109/TPAMI.2015.2389824*.

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

[40] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[42] W. Wang *et al.*, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[43] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[44] G. Ghiasi, T.-Y. Lin, and Q. V Le, "Nas-fpn: Learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 7036–7045.

[45] M. Tan, R. Pang, and Q. V Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10781–10790.

[46] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.

[47] K. He and G. Gkioxari, "P. Doll ar, and R. Girshick,'Mask r-CNN,"" in *Proc. IEEE Int. Conf. Comput. Vis*, 2017, pp. 2980–2988.

[48] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[49] K. Chen *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4974–4983.

[50] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[51] S. Qiao, L.-C. Chen, and A. Yuille, "Detectors: *the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2820–2828.

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004. *https://doi.org/10.1023/B:VISI.0000029664.99615.94*.

[29] T. Nguyen, E.-A. Park, J. Han, D.-C. Park, and S.-Y. Min, "Object detection using scale invariant feature transform," in *Genetic and Evolutionary Computing: Proceedings of the Seventh International Conference on Genetic and Evolutionary Computing, ICGEC 2013, August 25-27, 2013-Prague, Czech Republic*, 2014, pp. 65–72.

[30] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–361, 2001. *https://doi.org/ 10.1109/34.917571*.

[31] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004, vol. 2, pp. II--II.

[32] E. Hsiao, P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," 2009.

[33] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2009. *https://doi.org/10.1109/TPAMI.2009.167*.

[34] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *2010 IEEE Computer society conference on computer vision and pattern recognition*, 2010, pp. 2241–2248.

[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[36] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, pp. 154–171, 2013. *https://doi.org/10.1007/s11263-013-0620-5*.

[37] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989. *https://doi.org/10.1162/neco.1989.1.4.541*.

Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[64] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *Computer Vision--ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, 2016, pp. 483–499.

[65] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects,” *IEEE Trans. neural networks Learn. Syst.*, *https://doi.org/202110.1109/TNNLS.2021.3084827*.

[66] I. Loshchilov and F. Hutter, “Sgdr: Stochastic gradient descent with warm restarts,” *arXiv Prepr. arXiv1608.03983*, 2016.

[67] D. Misra, “Mish: A self regularized non-monotonic activation function,” *arXiv Prepr. arXiv1908.08681*, 2019.

[68] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, 2023. *https://doi.org/10.5555/3455716.3455856*

[69] 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, vol. 1, p. 2. *https://doi.org/10.48550/arXiv.1810.04805*.

[70] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, and others, “Improving language understanding by generative pre-training,” 2018.

[71] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020, *https://doi.org/10.5555/3455716.3455856*.

[72] Z. Liu *et al.*, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10012–10022.

[73] A. G. Howard *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv Prepr. arXiv1704.04861*, 2017, *https://doi.org/10.48550/arXiv.1704.04861*.

[74] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[75] A. Howard *et al.*, “Searching for mobilenetv3,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

Detecting objects with recursive feature pyramid and switchable atrous convolution,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10213–10224.

[52] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[53] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, “A real-time algorithm for signal analysis with the help of the wavelet transform,” in *Wavelets: Time-Frequency Methods and Phase Space Proceedings of the International Conference, Marseille, France, December 14--18, 1987*, 1990, pp. 286–297.

[54] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[55] 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.

[56] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[57] L. Zhao and S. Li, “Object detection algorithm based on improved YOLOv3,” *Electronics*, vol. 9, no. 3, p. 537, 2020, *https://doi.org/10.3390/electronics9030537*.

[58] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv Prepr. arXiv2004.10934*, 2020.

[59] O. E. Olorunshola, M. E. Irhebhude, and A. E. Evwiekpaefe, “A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms,” *J. Comput. Soc. Informatics*, vol. 2, no. 1, pp. 1–12, 2023, *https://doi.org/10.33736/jcsi.5070.2023*.

[60] C. Huanjie *et al.*, “SSD object detection algorithm with multi-scale convolution feature fusion,” *J. Front. Comput. Sci. Technol.*, vol. 13, no. 6, p. 1049, 2019.

[61] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, “Scalable object detection using deep neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2147–2154.

[62] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[63] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P.

[76] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[77] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient cnn architecture design,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 116–131.

[78] R. J. Wang, X. Li, and C. X. Ling, “Pelee: A real-time object detection system on mobile devices,” *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[79] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, “Dsod: Learning deeply supervised object detectors from scratch,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1919–1927.

[80] C. Liu *et al.*, “Progressive neural architecture search,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19–34.

[81] T.-J. Yang *et al.*, “Netadapt: Platform-aware neural network adaptation for mobile applications,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 285–300.