بسمه تعالی

مرور الگوریتم های تشخیص اشیا

ر.ز

مرداد ۱۴۰۱

چکیده

در این گزارش ابتدا به مرور اجمالی انواع الگوریتم های تشـــخیص اشـــیا پرداخته و با ارائه دســـته بندی های گوناگون ت صویری ن سبتا جامع از الگوریتم های مهم تر سیم می کنیم. د سته بندی ها بر مبناهای مختلفی ارائه شده اند مانند: کلاسیک-عمیق، معماری بک بون، یک/دو مرحله ای بودن، کانولوشنی-ترنسفورمر، نظارت کامل-ضعیف و دسته بندی بر اساس چالش های موجود در این حوزه آورده شده است.

سپس از میان موارد جدید یک مورد را انتخاب و پیاده سازی می کنیم. هم چنین تو ضیحاتی مخت صر در مورد نحوه ی ســتاپ برد جتســون و اشــتراک اینترنت (برای نصــب پکیج ها) روی برد در مد headless (بدون مانیتور اضافه و کابل شبکه) و نکاتی پیرامون دانلود تصاویر داکری آورده شده است.

مقدمه: انواع الگوریتم های تشخیص تصویر

در یک تقسیم بندی کلی ابتدا می توان این الگوریتم ها را به دو دسته ی کلاسیک و شبکه عصبی تقسیم کرد. در د سته ی اول الگوریتم هایی مانند Viola Johns (2001), HOG (2006), DPM (2008) قرار دارند. از سال ۲۰۱۴ الگوریتم های مبتنی بر یادگیری عمیق به سرعت گسترش یافتند که به طور عمده به دو دسته ی یک و دو مرحله ای (استیج) تقسیم می شوند. در مدل های دو مرحله ای ابتدا تعداد اجسام و ناحیه ی مورد علاقه در فریم تخمین زده شده و سپس اجسام دسته بندی شده و برای آن ها کادر (boundary box regression) ترسیم می شود. مانند:

RCNN/SPPNet 2014), Fast RCNN (2015), Mask RCNN (2017), Pyramid Net/FPN (2017), GRCNN (2021).

برخی مدل ها برای رسیدن به سرعت بالاتر دو مرحله را ادغام می کنند، مانند:

YOLO (2016), SSD (2016), RetinaNet (2017), YOLOv3(2018) ,YOLOv4 (2020), YOLOR (2021), YOLOv5 (2021).

هم چنین از دیدگاه دیگر میتوان این مدل ها بر حسب backbone آن ها دسته بندی کرد. دسته بندی بر اساس مدل های سنگین و سبک (مناسب ادوات اج) و هم چنین با دیدگاه یادگیری با نظارت ضعیف نیز به مساله نگاه شده است. از آن جایی که هر اصلاح و توسعه ای برای رفع چالشی می باشد، دسته بندی بر اساس چالش ها نیز آورده شده است.

انواع Backbone در مدل های تشخیص اشیا

هر دو نوع مدل تک و دو مرحله ای، دارای دو بخش backbone و head network می با شند. بخش backbone دارای وظیفه ی ا ستخراج ویژگی از ت صاویر و بخش head برای محل یابی و ر سم bbox و همچنین طبقه بندی است. در شکل ۱ این دو بخش برای هر دو دسته ی اصلی الگوریتم های تشخیص اشیا مشاهده می شود.

لذا می توان از زاویه دید backbone به کار رفته در معماری به مدل نگریست و به تقسیم بندی شبکه های مختلف پرداخت. در این بخش چند backbone معروف ارائه می شود. مرجع اصلی این بخش (Zaidi, 2021) می باشد.

AlexNet

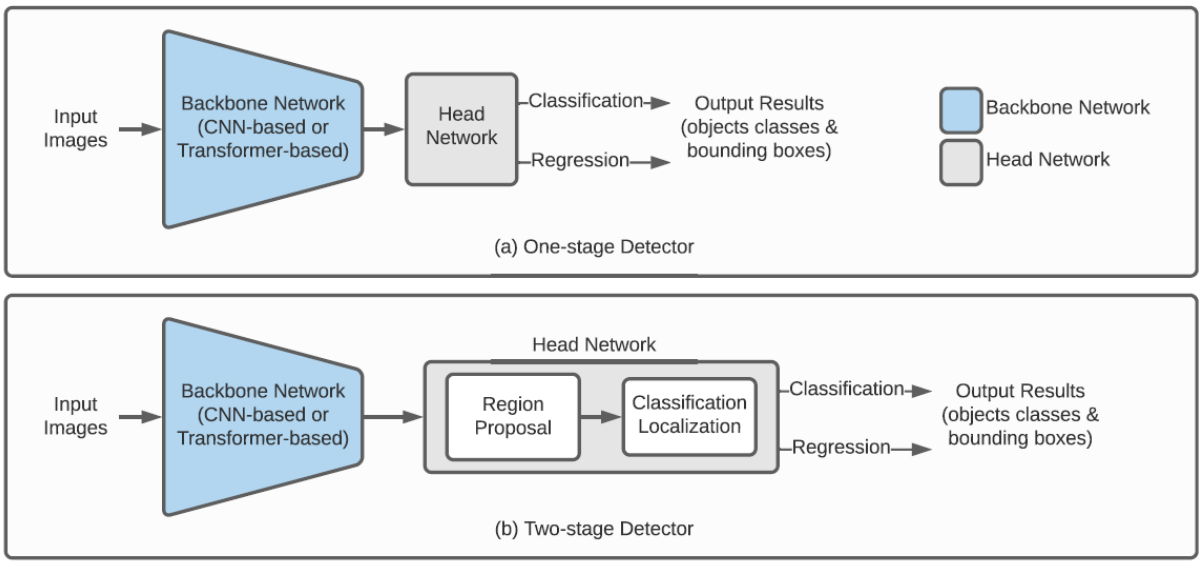
این مدل تو سط Krizhevsky در سال ۲۰۱۲ و بر مبنای شبکه ی ع صبی کانولو شنی (CNN) ارائه و برنده ی جایزه ی ILSVRC2012 شد. دارای ه شت لایه یادگیرنده (پنج لایه کانولو شنی و سه لایه ی فول کانکت) و یک لایه ی سافت مکس در انتهای خود است. هم چنین از تابع فعالساز ReLu و ایده ی dropout برای رگولازیسیون بهره می برد.

VGG

این شبکه در ادامه AlexNet با عمیق تر کردن و افزایش تعداد لایه ها تا ۱۶-۱۹ لایه، سعی در بهبود دقت دارد. برای مدیریت ناپایداری گرادیان ابتدا یازده لایه از شبکه آموزش دیده و وزن های بدست آمده به عنوان نقطه شروع برای آموزش کل شبکه استفاده می شود.

GoogLeNet/Inception

تعداد پارامترهای زیاد در شبکه منجر به محاسبات بیشتر و اسیب پذیری نسبت به اورفیت می شود. برای دوری از این م سایل پی شنهاد ا ستفاده از لایه های به طور محلی تنک (ا سپارس) بجای لایه تماما مت صل داده شد. به علاوه GoogleNet از ماژول inception که متشکل از فیلترهایی با سایزهای گوناگون می باشد استفاده می کند.



شکل ۱ بخش backbone و head برای هر دو نوع مدل تک(a) و دو مرحله ای(b)

ResNet

این شبکه برای ا ستفاده از لایه های بی شتر از ایده ی skip connection و ا ضافه کردن یک ات صال ساده بدون افزودن پارامتر و پیچیدگی به شبکه بهره می برد.به طوری که شبکه ی VGG با ۱۶ لایه از این شبکه با ۱۰۱ لایه پیچیده تر است. در ورژن دوم این معماری از ReLu و Batch Normalization استفاده شده است.

ResNeXt نیز با ا ستفاده از ماژول های inception و تعبیه چند م سیری در بلوک مدل را برای دقت بی شتر توسعه داد.

CSPNet: Cross Stage Partial Network

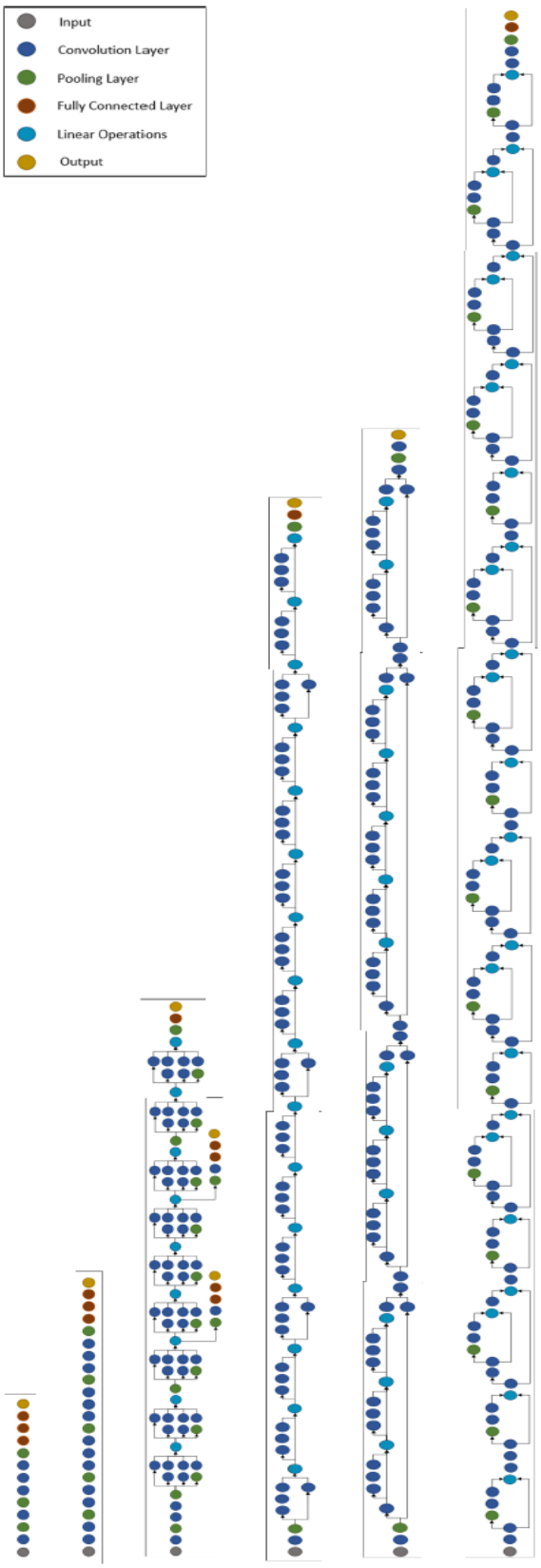
ایده ی ا صلی حذف اطلاعات گرادیان های تکراری از شبکه با ایجاد دو م سیر برای گرادیان می با شد. با تق سیم لایه بیس به دو بخش و سپس ات صال یک بخش آن به لایه های کانولو شنی و بخش دیگر با خروجی در مرحله ی بعد ترکیب می شود. این ایده می تواند به شبکه های مختلف اعمال شود و محا سبات را ۱۰ الی ۲۰ در صد کاهش دهد.

EfficientNet

در صورتی که توان محا سباتی بی شتری در د سترس ا ست میتوان با تغییر پارامتر .compound coef به طور یکنواخت شبکه را در راستای طول، عرض و عمق گسترش داد.

Model	Year	Layers	Parameters (Million)	Top-1 acc%	FLOPs (Billion)
AlexNet	2012	7	62.4	63.3	1.5
VGG-16	2014	16	138.4	73	15.5
GoogLeNet	2014	22	6.7	-	1.6
ResNet-50	2015	50	25.6	76	3.8
ResNeXt-50	2016	50	25	77.8	4.2
CSPResNeXt-50	2019	59	20.5	78.2	7.9
EfficientNet-B4	2019	160	19	83	4.2

شکل ۲ جدول مقایسه ی پارامتری شبکه های معروف بکار رفته به عنوان backbone



شکل ۳ بصری سازی معماری شبکه های (از راست به چپ): *Efficient net-B4, CSPResNetXt, ResNet-50, GoogleNet, VGG-16, AlexNet*

الگوریتم های کلاسیک تشخیص تصویر

الگوریتم Viola Jones

به طور خلاصـــه، برای فیلتر باکس بهتر از روش انتگرال تصـــویر، برای نمایش ویژگی از وولت هار، برای انتخاب ویژگی از آدابوست و برای حذف پس زمینه از طبقه بند آبشاری استفاده می کند.

الگوریتم HOG: Histogram of Oriented Gradient

با اســـتفاده از الگوریتم Sobel لبه یابی روی پچ هایی با نســـبت ابعاد ۲:۱ انجام شـــده و ســـپس بر مبنای جهت گرادیان، هیســـتوگرام بدســـت می آید. پس از SVM نیز از non max suppression (NMS) برای طبقه بندی صحیح استفاده می شود. HOG به طور گسترده برای تشخیص افراد پیاده استفاده می شود.

الگوریتم DPM: Deformable Part-based Model

DPM را می توان تو سعه HOG در نظر گرفت که از ا ستراتژی "تق سیم کن و حکومت کن" ا ستفاده می کند. یعنی در فاز آموزش شی مورد نظر به اجزای خود افراز شده و پس از بررسی هر جز، با ادغام نتایج، تفسیر نهایی ارائه می گردد.

از آنجایی که تاکید ا صلی این گزارش بر روی الگوریتم های جدید و کارآمد ما شین لرنینگ می با شد، به همین مقدار بسنده کرده و در بخش های بعدی به ارائه ی تقسیم بندی های گوناگون مدل های تشخیص ا شیا مبتنی بر شبکه عصبی می پردازیم. مرجع اصلی مدل های تک و دو مرحله ای (Zaidi, 2021) می باشد.

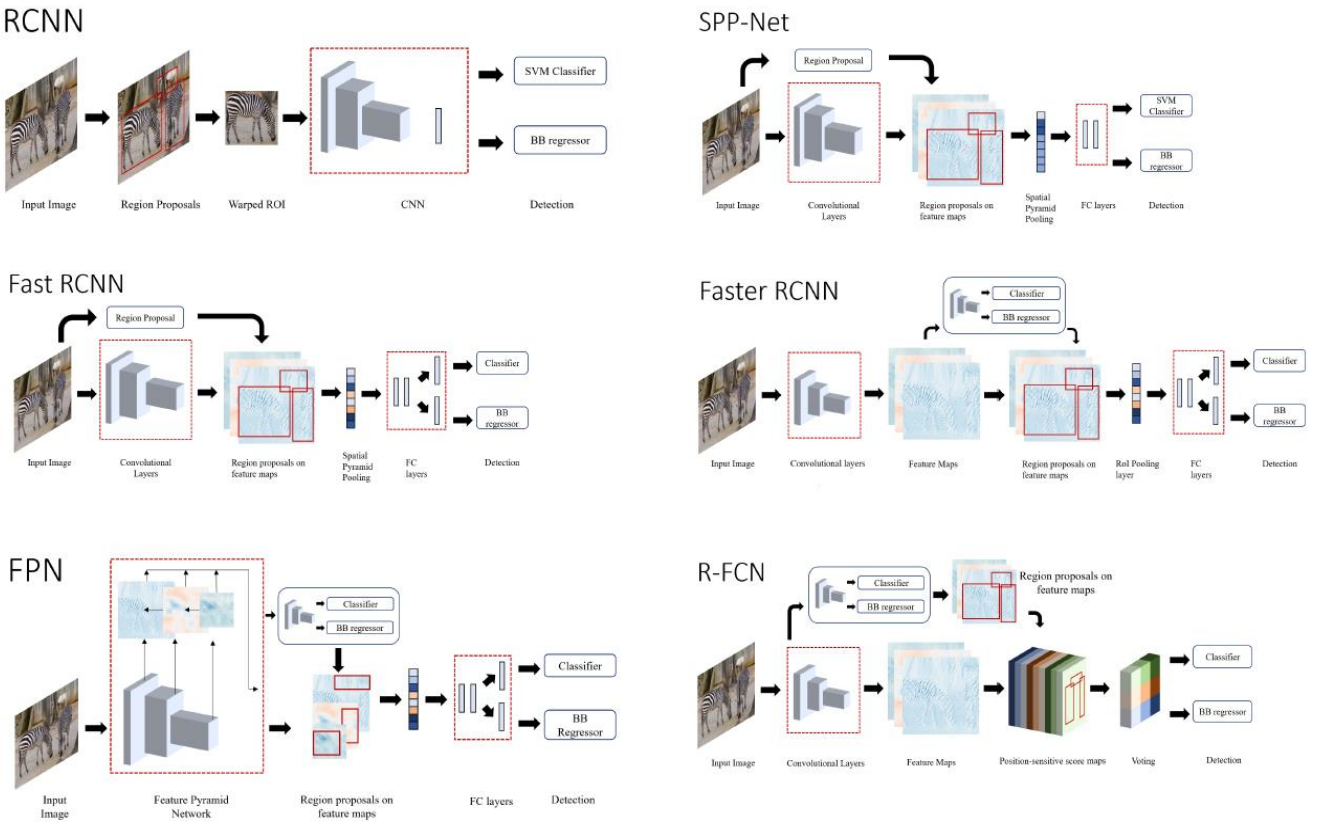
الگوریتم های تشخیص تصویر مبتنی بر شبکه عصبی: الف) Two Stage

Region based CNN (RCNN)

در مرحله ی ا ستخراج ناحیه، با ا ستفاده از الگوریتم selective search ، دوهزار تصویر در مقیاس های گوناگون جدا شده و سپس بر مبنای رنگ، اندازه، بافت و شکل به چند ناحیه ی اصلی ادغام می شوند. سپس هر ناحیه به سایز ثابتی تغیر اندازه داده و به شبکه CNN برای استخراج ویژگی ارسال می شود. برای طبقه بندی از SVM و سپس NMS با معیار IOU استفاده می شود. معماری این مدل در شکل ۵ قابل مشاهده است.

Spatial Pyramid Polling (SPP)

برای حذف محدودیت ثابت بودن ابعاد تصویر ورودی در RCNN می توان از SPP استفاده کرد تا دیگر مجبور به crop/wrap کردن تصاویر نباشیم و هم چنین سرعت مدل حدود بیست برابر بیشتر می شود (شکل ۶).



شکل ٤ معماری شبکه های معروف تشخیص اشیا دو مرحله ای

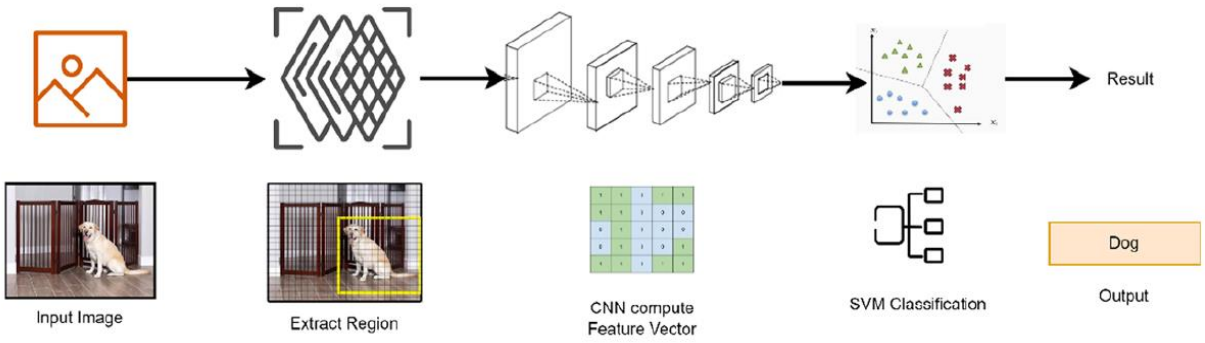
Fast RCNN

این مدل با استفاده از RPN: Regional Proposal Net. (شکل ۸) که از sliding window روی نقشه ی ویژگی برای تولید bbox (anchor) اســتفاده میکند، ســعی در بهبود RCNN دارد. در انتها نیز از softmax برای تعیین

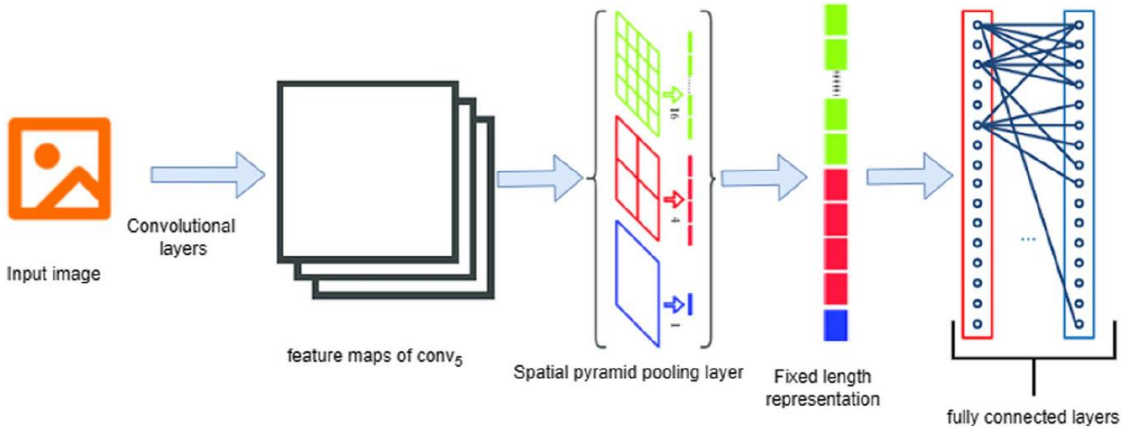
احتمال هر کلاس و از رگرسیون برای تعیین bbox استفاده می شود. هم چنین مدل MASK RCNN -که برای سگمنت تصویر (کلاس بندی در سطح پیکسل) مورد استفاده قرار می گیرد- را می توان توسعه ای از مدل Fast RCNN دانست. در شکل ۷ معماری این مدل مشاهده می شود.

DetectoRS

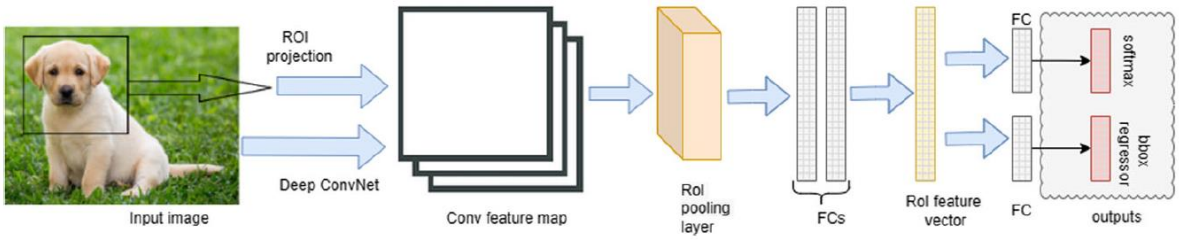
در این مدل، دیدگاه دو مرحله ای در دو سطح میکرو و ماکرو اعمال می شود. در سطح ماکرو از RFP و FPN چند گانه به همراه اتصالات فیدبک استفاده شده و خروجی FPN تو سط ASPP پردازش می شود. در سطح میکرو نیز از Switchable Atrous Convolution استفاده می شود.



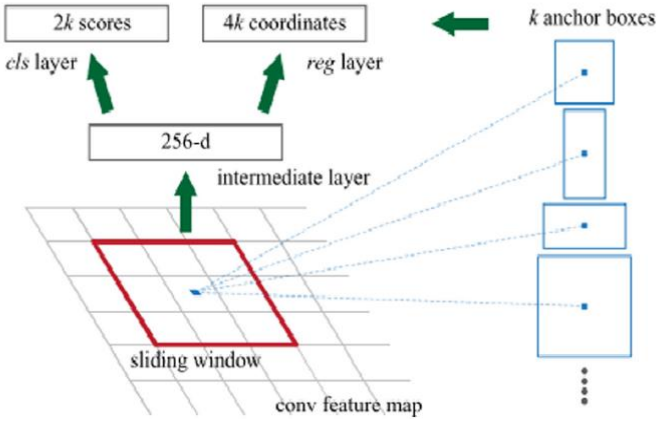
شکل ۵ ساختار الگوریتم RCNN: استخراج نواحی، استخراج ویژگی توسط CNN و SVM



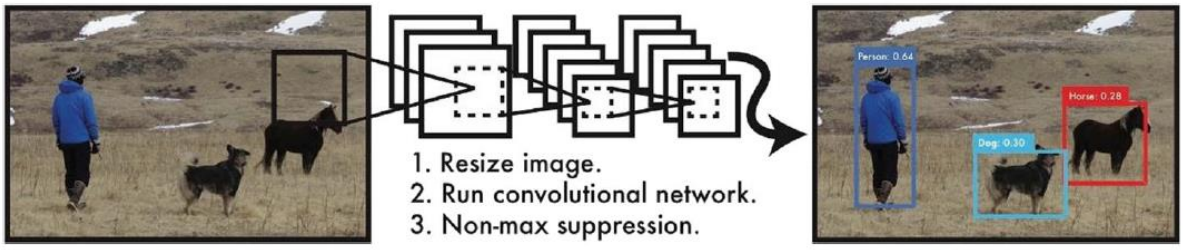
شکل ٦ استفاده از لایه SPP برای حذف مرحله ی برش و تغییر سایزبه عدد ثابت



شکل ۷ساختار fast RCNN



شکل ۸ ساختار RPN: Region Proposal Network



شکل ۹ نمایی از شبکه ی یولو

الگوریتم های تشخیص تصویر مبتنی بر شبکه عصبی: ب) One Stage

YOLO: You Only Lock Once

در این مدل تصویر به نواحی با سایز ثابت تقسیم می شود و در هر ناحیه احتمال و bbox کلاس محاسبه می شود. در ورژن دوم از Darknet-19 به عنوان backbone استفاده می شود. هم چنین از نرمالیزاسیون بچ برای جلوگیری از اورفیتینگ بهره می برد. در ورژن سوم Darknet-53 و طبقه بند لجستیک استفاده شده است. هم چنین با استفاده از FPN: Feature Pyramid Network یادگیری در مقیاس های گوناگون صورت می پذیرد.

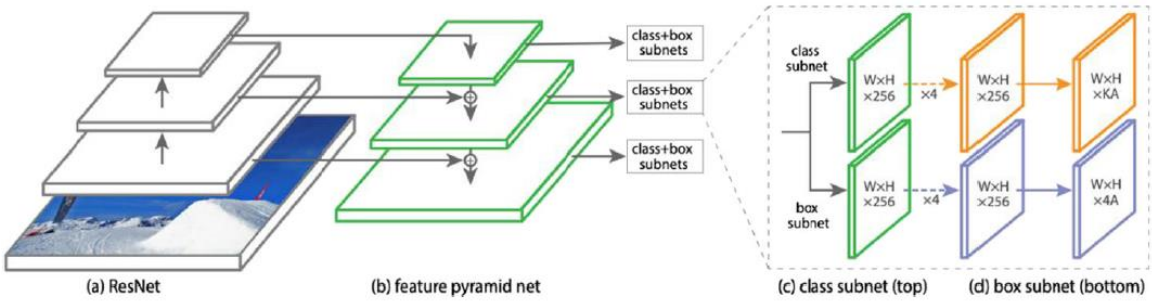
SSD: Single Shot multibox Detector

این مدل برای افزایش دقت در شناسایی اجسام کوچک از نمایش چند مقیاسی استفاده می کند. ورژن deconvolution SSD را می توان مدل توسعه یافته RCNN که از ResNeT برای backbone استفاده می کند، دانست.

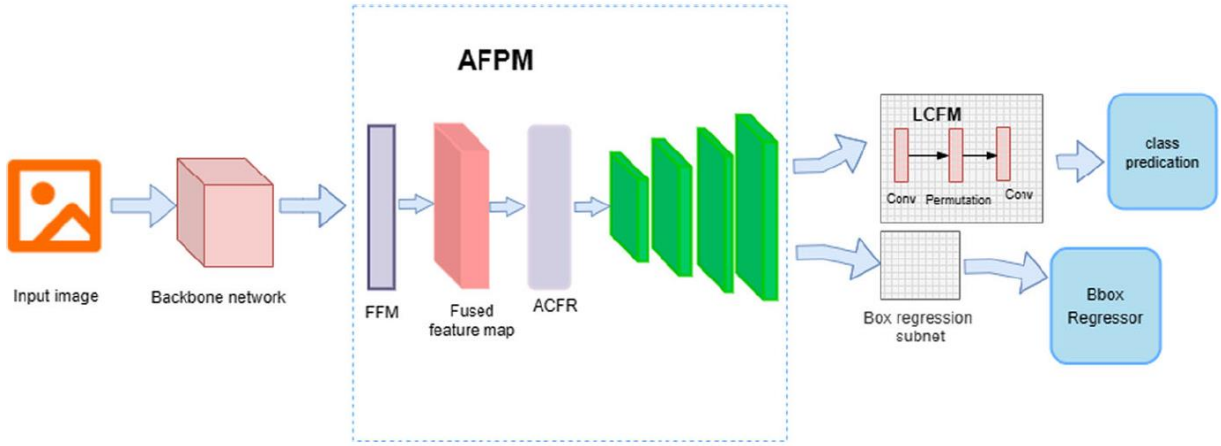
Retina Net

این مدل یک تابع لاس جدید با نام Focal loss برای رفع مشکل عدم توازن تعداد اعضای هر کلاس در مرحله ی آموزش معرفی کرده است. این تابع با اضافه کردن یک ترم دینامیکی، تابع لاس آنتروپی رو به گونه ای مدوله می کند تا شبکه بیشتر بر روی مثال های سخت متمرکز شود.

برای افزایش دقت بخصوص برای اجسام کوچک از رهیافت Feature Pyramid Net: FPN استفاده می کند تا بتواند اجسام در مقیاس های گوناگون را شناسایی کند. از دیدگاهی دارای دو زیر شبکه برای طبقه بندی و رگرسیون bbox دارد و از طرفی دو مسیر پایین به بالا (کاهش ابعاد تصویر طی کانلولوشن های متوالی با گام دو) و هم چنین مسیر بالا به پایین برای افزایش رزولوشن نقشه ویژگی بدست آمده پیاده سازی شده است. طبقه بند و رگرسیون bbox به هر طبقه اعمال می شود.



شکل ۱۰ ساختار شبکه Retina که از FPN بهره می برد



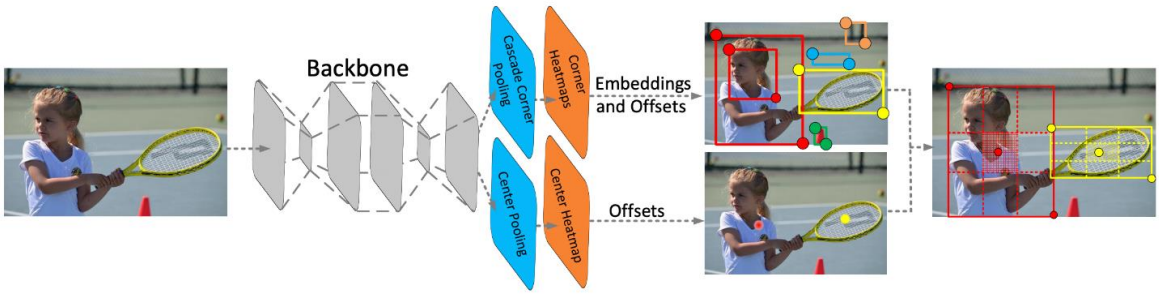
شکل ۱۱ ساختار شبکه LADet

Lightweight and Adaptive Network for Multi-scale Object Detection: LADet

این مدل دارای backbone دارک نت ۱۶۹ بوده و دو ماژول اصلی دارد: Adaptive Feature Pyramid Module و Light-weight Classification Function Module. همان طور که از اسم مدل مشخص است تمرکز اصلی بر روی مدیریت چالش تغییر مقیاس می باشد.

CenterNet

یک رهیافت متفاوت از نمایش نتایج تو سط bbox اتخاذ کرده ا ست. تصویر وارد یک FCN شده و خروجی یک heatmap است که بیشینه آن نشانگر مرکز شی تشخیص داده شده است. استخراجگر ویژگی این مدل ایمیجنت از پیش آموزش دیده به همراه Hourglass-101 می با شد. سپس سه head برای تشخیص مرکز، ابعاد و آفست ا ستفاده می شود. Non max suppression نیز دیگر لازم نیست.این شبکه دقت خوبی روی تسک هایی مانند شناسایی اجسام سه بعدی، سگمنتیشن و تشخیص جهت داشته است.



شکل ۱۲ ساختار شبکه ی CenterNet

الگوریتم های تشخیص اشیا مبتنی بر ترنسفورمر

در یک تقسیم بندی دیگر می توان مدل های تشخیص اشیا را به دو دسته ی مبتنی بر شبکه ی کانولوشنی و مبتنی بر ترنسفورمر تقسیم کرد. در این دیدگاه تقریبا تمامی شبکه های معرفی شده جز دسته ی کانولوشنی هستند بجز شبکه های جدیدی مانند Swin که در این بخش معرفی می شود. این شبکه از دیدگاه تک یا دو مرحله ای بودن، در دسته ی تک مرحله ای قرار می گیرد.

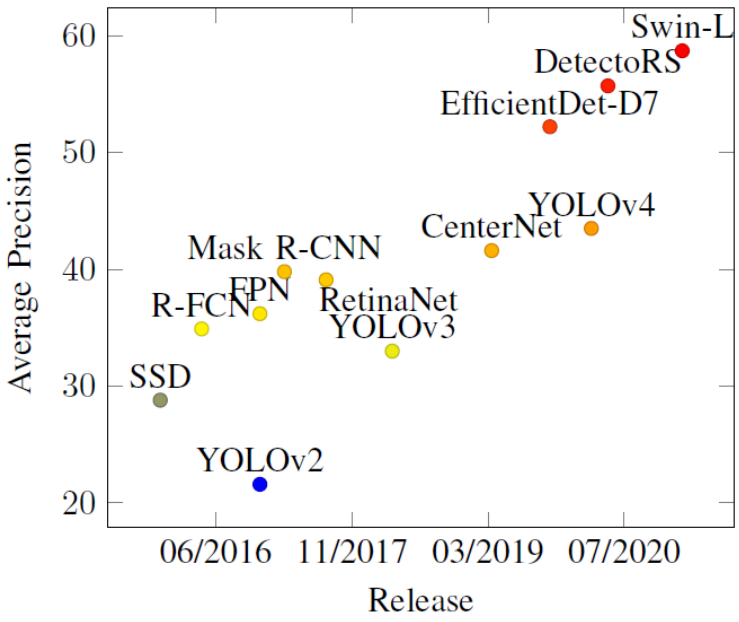
Swin Transformer

ترنسفورمرها با ورود مکانیزم توجه توانستن نقطه ی عطفی در حوزه ی پردازش زبان طبیعی بوجود بیاورند. مدل های معروفی مانند BERT, GPT-3, T5 از ترنسفورمر استفاده می کنند. Swin یک backbone برای تسک های حوزه ی بینایی ماشین می باشد که بجای شبکه های کانولوشنی بر ترنسفورمرها مبتنی است. ترنسفورمر سوین ابتدا تصویر ورودی را به پچ های غیر همپوشان تقسیم کرده و به امبدینگ تبدیل می کند. تعداد زیادی بلاک های سوین ترنسفورمر بر روی این واحد ها اعمال شده و در چهار مرحله کاهش ابعاد صورت می پذیرد تا به ساختاری سلسله مراتبی دست یابد. هر کدام ازین بلوک ها از ماژول local multi-headed self-attention (MSA) تشکیل شده است. با مکانیزم شیفت پنجره می توان به محاسبات متناسب با سایز تصویر (بجای ابعاد به توان دو) رسید.

جدول ۱ جدول مقایسه ی مدل های گوناگون بر حسب سال ارائه، بک بون، اندازه، دقت و سرعت

Model	Year	Backbone	Size	$AP_{[0.5:0.95]}$	$AP_{0.5}$	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%	~3
FPN	2017	ResNet-101	800	36.20%	59.10%	5
Mask R-CNN	2018	ResNeXt-101-FPN	800	39.80%	62.30%	5
DetectoRS	2020	ResNeXt-101	1333	53.30%	71.60%	~4
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
YOLOv4	2020	CSPDarkNet-53	512	43.00%	64.90%	31
Swin-L	2021	HTC++	-	57.70%	-	-

[a]Models marked with * are compared on PASCAL VOC 2012, while others on MS COCO.Rows colored gray are real-time detectors (>30 FPS).



شکل ۱۳ دقت الگوریتم ها (AP) و سال انتشار برای مدل های معروف اخیر

الگوریتم های سبک (lightweight)

از دیدگاهی دیگر می توان الگوریتم های تشخیص اشیا را به دو دسته سبک و غیر سبک تقسیم کرد. از آنجایی که مدل هایی که تا کنون برر سی شدند عمدتا جنرال و غیر سبک بودند. البته برخی از آن ها ورژن سبک هم دارند، مانند YOLOv5n و هم چنین برخی روش های مخصوص سبک سازی شبکه مانند کوانتیزاسیون، هرس کردن و تقطیر دانش نیز وجود دارد. در این بخش به مدل هایی که مخ صو صا با حجم کم و برای ادوات با قدرت محاسباتی کم (مانند موبایل یا بردهای امبدد) طراحی شده اند، معرفی می شوند.

SqueezeNet

ماژول های سازنده ی این شبکه fire نام دارند که دارای دو لایه ی squeeze (شامل فیلترهای 1x1) و expand (شـامل فیلتر های 1x1, 3x3) می باشـند. هر دو لایه از تابع فعالسـاز ReLu اسـتفاد می کنند. هم چنین ورژن دارای اتصال residual نیز ازین مدل برای افزایش دقت توسعه یافته است.

Mobilenet

در این شبکه به جای ا ستفاده از لایه های کانولو شنی معمولی از ۲۸ لایه depthwise separable convolution اسـتفاده شـده اسـت. این رهیافت هزینه ی محاسـباتی و سـایز مدل را کاهش می دهد. هم چنین از ReLu به همراه batch norm نیز بهره می برد.

در سال ۲۰۱۸ ورژن دوم این معماری با معرفی لایه ای جدید به نام inverted residual with linear bottleneck ارائه شد. ابتدا نمایش ویژگی با بعد کم به ابعاد بالاتر گسترش می یابد، پس از فیلتر شدن با depthwise conv. به ابعاد پایینتر دوباره بر می گردد. هم چنین از ReLu6 بجای ReLu استفاده می کند.

در ورژن سوم نیز از جستجوی اتوماتیک معماری با NetAdapt استفاده شده است. هم چنین Howard این ایده را مطرح کرد که فیلترها خیلی اوقات تصویر آینه ای از همدیگرند و لذا با حذف نیمی از آن ها می توان دقت را تا حد زیادی حفظ کرد. در ورژن سوم از ترکیب swish hard و ReLu برای تابع فعال ساز استفاده شده است.

ShuffleNet

به طور خاص برای موبایل در سال ۲۰۱۷ ارائه شد. بلوک های شافلنت مانند بلوک های رزنت هستند که از depthwise conv. و بجای 1x1 conv. از pointwise group conv. استفاده می کنند. این روش ها گرچه منجر به کاهش سایز مدل می شود اما در عمل تاثیری روی زمان اینفرنس ندارد. در سال ۲۰۱۸ ورژن دوم شافل نت با تاکید بیشتر بر روی متریک های مستقیم مانند سرعت و لتنسی به جای متریک های غیر مستقیم مانند FLOP ارائه شد. کانال های ورودی و خروجی دارای عرض یکسان هستند تا هزینه ی دسترسی به حافظه کاهش یابد. هم چنین از ساختار چند مسیره و اپراتور های element wise استفاده شده است.

Mansnet

در سال ۲۰۱۸ و با روش Neural Architecture Search (NAS) اتوماتیک مساله را تبدیل به بهینه سازی چند هدفه (افزایش دقت و کاهش تاخیر) کرده و با ملاحظاتی فضای سرچ را محدود کردند و سپس از عامل یادگیری تقویتی بر مبنای RNN استفاده کردند. مدل بدست آمده دو برابر موبایل نت۲ سرعت دارد.

OFA: One For ALL

ا ستفاده از رهیافت جستجوی اتوماتیک معماری شبکه (NAS) توان محا سباتی بالایی نیاز دارد. Cai یک روش نوآورانه برای حل این چالش ارائه داد. ابتدا تمامی هایپرپارامتر های شبکه در ماکسیمم خود قرار داده می شوند تا بزرگترین شبکه در فضای جستجو ساخته شود. سپس یک بار این شبکه بزرگ آموزش می بیند. پس از این جستجو در فضای سرچ آغاز شده و تدریجا شبکه کوچک می شود و هر بار صرفا fine tune انجام می شود. این ایده برنده ی مسابقه LPCVC با دقت ۸۰ درصد و چندین مرتبه کاهش در هزینه محاسباتی شد.

جدول2 مقایسه مدل های سبک بر حسب سال، دقت، سرعت، پارامتر و بار محاسباتی

Model	Year	Top-1 Acc%	Latency (ms)	Parameters (Million)	FLOPs (Million)
SqueezeNet	2016	60.5	-	3.2	833
MobileNet	2017	70.6	113	4.2	569
ShuffleNet	2017	73.3	108	5.4	524
MobileNetv2	2018	74.7	143	6.9	300
PeleeNet	2018	72.6	-	2.8	508
ShuffleNetv2	2018	75.4	178	7.4	597
MnasNet	2018	76.7	103	5.2	403
MobileNetv3	2019	75.2	58	5.4	219
OFA	2020	80.0	58	7.7	595

تشخیص اشیا با نظارت ضعیف WSOD

در تقسیم بندی دیگری میتوان الگوریتم های تشخیص تصویر را به دو دسته ی نظارت شده ی کامل و نظارت شده ی ضعیف[1] تقسیم کرد. در نوع ضعیف لیبل گذاری در سطح تصویر انجام می شود، یعنی مشخص می شود که این تصویر شامل چه کلاس هایی می باشد. یکی از مهمترین مزیت های این نوع رهیافت، امکان استفاده از دادگان ارزان تر و بیشتر می باشد. مرجع این بخش مقاله (Shao, 2021) می باشد.

به طور عمده می توان این الگوریتم ها را به دو دسته تقسیم کرد:

الف) چند نمونه در تصویر multiple instance in image (MIL) که داری سه بخش اصلی می باشد:

Proposal generator بخاطر سرعت بیشتر SW: sliding window نسبت به SS: selective search و EB: edge box ، مرسوم تر است.

Backbone استفاده از مواردی مانند GoogleNet, ResNet, SeNet, VGG16 هم برای کلاس بندی و هم برای تشخیص اشیا مرسوم است

Detection head دارای دو جریان مکان یابی و طبقه بندی

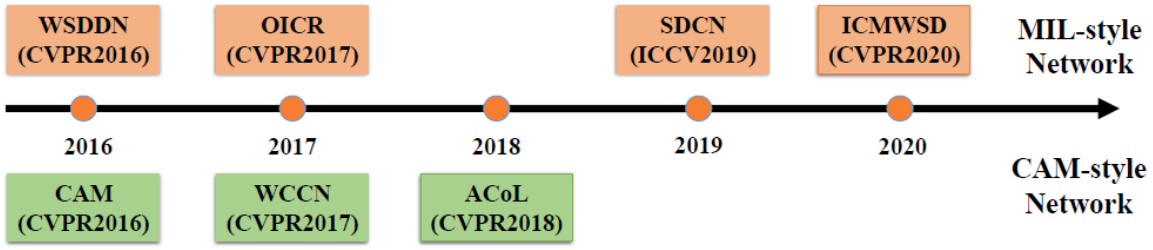
[1] WSOD: weakly supervised object detection

ب) شبکه بر مبنای CAM: دارای سه بخش اصلی:

Classifier: شامل لایه کانولوشنی، FN و global average polling

Backbone: مشابه MIL

Class activation map: وظیفه ی مکان یابی توسط تکنیک ساده ی سگمنتیشن



شکل ۱۴ شبکه های به کار رفته برای تشخیص اشیا با نظارت ضعیف

مرجع (Shao, 2021) در ادامه به تو ضیح چندین روش و هم چنین راه حل های آنان برای چالش های WSOD پرداخته است در اینجا جداول نهایی مرتبط با این گزارش آورده می شود:

جدول3 خلاصه ای از شبکه های WSOD به همراه اطلاعات سال انتشار کد (در صورت دسترسی) و پروپوزال (sw: sliding window, EB: edge box, SS: Selective search)

Approach	Year	Proposals	Network		Challenges			Code on Github
			MIL-based	CAM-based	Discriminative Region	Multiple Instances	Speed	
WSDDN [25]	CVPR2016	EB	√					hbilen/WSDDN
CAM [26]	CVPR2016	Heatmap		√			√	zhoubolei/CAM
WSLPDA [27]	CVPR2016	EB	√		√			jbhuang0604/WSL
WELDON [28]	CVPR2016	SW	√		√		√	
ContextLocNet [29]	ECCV2016	SS	√		√			vadimkantorov/contextlocnet
Grad-CAM [30]	ICCV2017	Heatmap		√	√		√	ramprs/grad-cam
OICR [31]	CVPR2017	SS	√		√			ppengtang/oicr
WCCN [32]	CVPR2017	EB	√		√			
ST-WSL [33]	CVPR2017	EB	√		√	√		
WILDCAT [34]	CVPR2017	Heatmap		√	√		√	durandtibo/wildcat.pytorch
SPN [35]	ICCV2017	SW	√				√	ZhouYanzhao/SPN
TP-WSL [36]	ICCV2017	Heatmap		√	√		√	
PCL [37]	TPAMI2018	SS	√		√	√		ppengtang/pcl.pytorch
GAL-fWSD [38]	CVPR2018	EB	√				√	
W2F [39]	CVPR2018	SS	√		√	√	√	
ACoL [40]	CVPR2018	Heatmap		√	√		√	xiaomengyc/ACoL
ZLDN [41]	CVPR2018	EB	√		√			
TS^2C [42]	ECCV2018	SS	√		√			
SPG [43]	ECCV2018	Heatmap		√			√	xiaomengyc/SPG
WSRPN [44]	ECCV2018	EB	√					
C-MIL [45]	CVPR2019	SS	√					WanFang13/C-MIL
WS-JDS [46]	CVPR2019	EB	√		√			shenyunhang/WS-JDS
ADL [47]	CVPR2019	Heatmap		√			√	junsukchoe/ADL
Pred NET [48]	CVPR2019	SS	√					
WSOD2 [49]	ICCV2019	SS	√		√			researchmm/WSOD2
OAILWSD [50]	ICCV2019	SS	√		√			
TPWSD [51]	ICCV2019	SS	√		√			
SDCN [52]	ICCV2019	SS	√		√			
C-MIDN [53]	ICCV2019	SS	√		√			
DANet [54]	ICCV2019	Heatmap		√			√	xuehaolan/DANet
NL-CCAM [55]	WACV2020	Heatmap		√	√		√	Yangseung/NL-CCAM
ICMWSD [23]	CVPR2020	SS	√		√			
EIL [56]	CVPR2020	Heatmap		√	√		√	Wayne-Mai/EIL
SLV [57]	CVPR2020	SS	√		√			

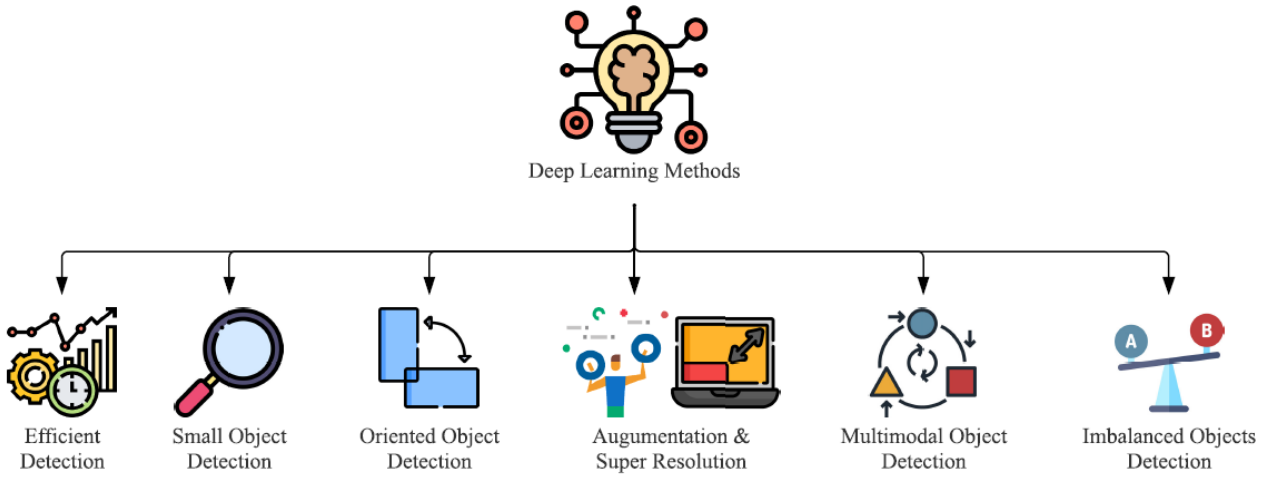
جدول 4 برخی تکنیک های مورد استفاده برای بهبود دقت شناسایی

Approach	Specific techniques for discriminative region problem								Training tricks		
	Cont	Self-t	Casc	BboxR	DisRR	Low-l	Seg-D	Trans	E-t-h	NegE	SmoL
WSDDN [25]											
CAM [26]											
WSLPDA [27]	√										
WELDON [28]										√	
ContextLocNet [29]	√										
Grad-CAM [30]						√					
OICR [31]		√						√			
WCCN [32]			√								
ST-WSL [33]		√						√			
WILDCAT [34]										√	
SPN [35]											
TP-WSL [36]					√						
PCL [37]		√						√			
GAL-fWSD [38]								√			
W2F [39]		√						√			
ACoL [40]					√						
ZLDN [41]									√		
TS^2C [42]	√		√					√			
SPG [43]											
WSRPN [44]											
C-MIL [45]											√
WS-JDS [46]							√	√			
ADL [47]											
Pred NET [48]				√				√			
WSOD2 [49]		√		√		√					
OAILWSD [50]	√	√									
TPWSD [51]		√		√							
SDCN [52]							√	√			
C-MIDN [53]		√			√			√			
DANet [54]											
NL-CCAM [55]										√	
ICMWSD [23]	√	√		√							
EIL [56]					√						
SLV [57]		√		√				√			

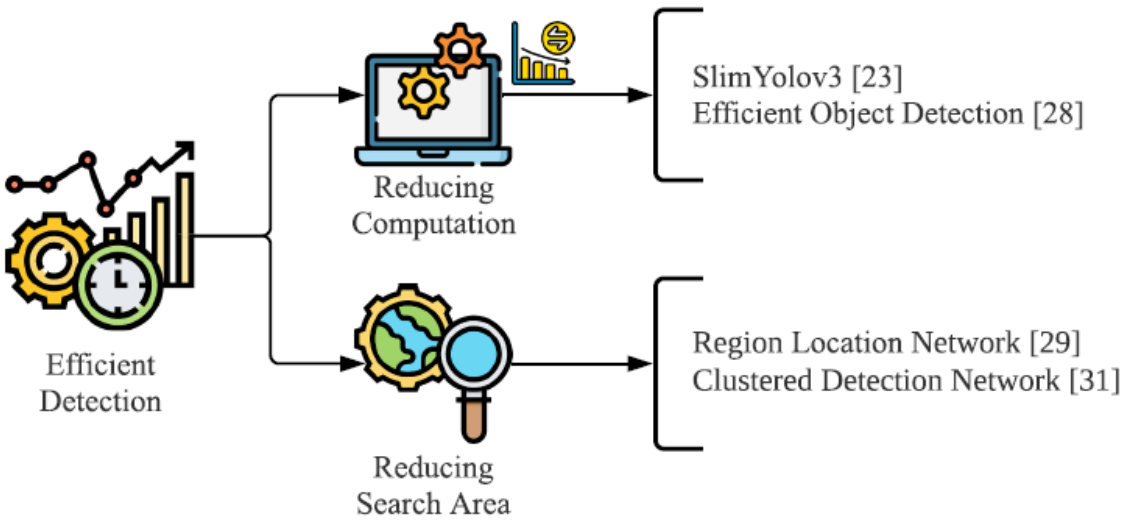
1) CONT: CONTEXT MODELING, 2) SELF-T: SELF-TRAINING ALGORITHM, 3) CASC: CASCADED NETWORK, 4) BBOXR: BOUNDING BOX REGRESSION, 5) DISRR: DISCRIMINATIVE REGION REMOVAL, 6) LOW-L: INCORPORATING LOW-LEVEL FEATURES, 7) SEG-D: SEGMENTATION-DETECTION COLLABORATIVE MECHANISM, 8) TRANS:TRANSFORMING WSOD TO FSOD, 9) E-T-H: EASY-TO-HARD STRATEGY, 10) NEGE: NEGATIVE EVIDENCE, 11) SMOL: OPTIMIZING SMOOTHED LOSS FUNCTIONS.

تقسیم بندی بر اساس چالش های موجود

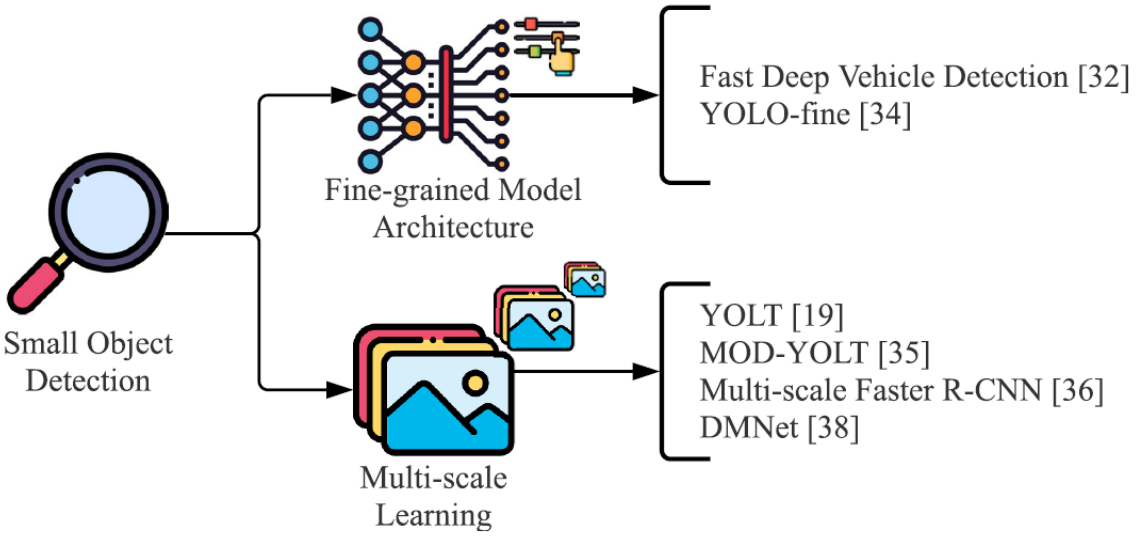
در مرجع (Kang, 2022) بر اساس چالش های موجود تقسیم بندی ارائه شده است. از آنجایی که تمرکز اصلی در این مرجع بر روی تصاویر هوایی بوده است به آوردن تصاویر به همراه توضیحات مختصر زیر هر شکل کفایت می کنیم:



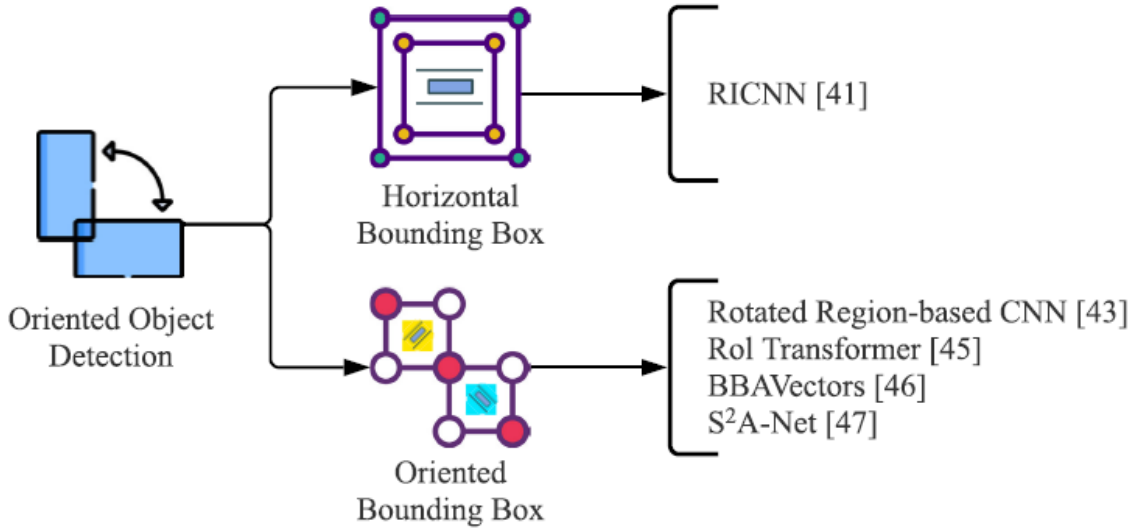
شکل ۱۵ تقسیم بندی روش های تشخیص اشیا بر مبنای چالش های شش گانه: شناسایی کارا، تشخیص اشیا کوچک، اشیا جهت دار، رزولوشن بالا، چندمدالیته و عدم تعادل



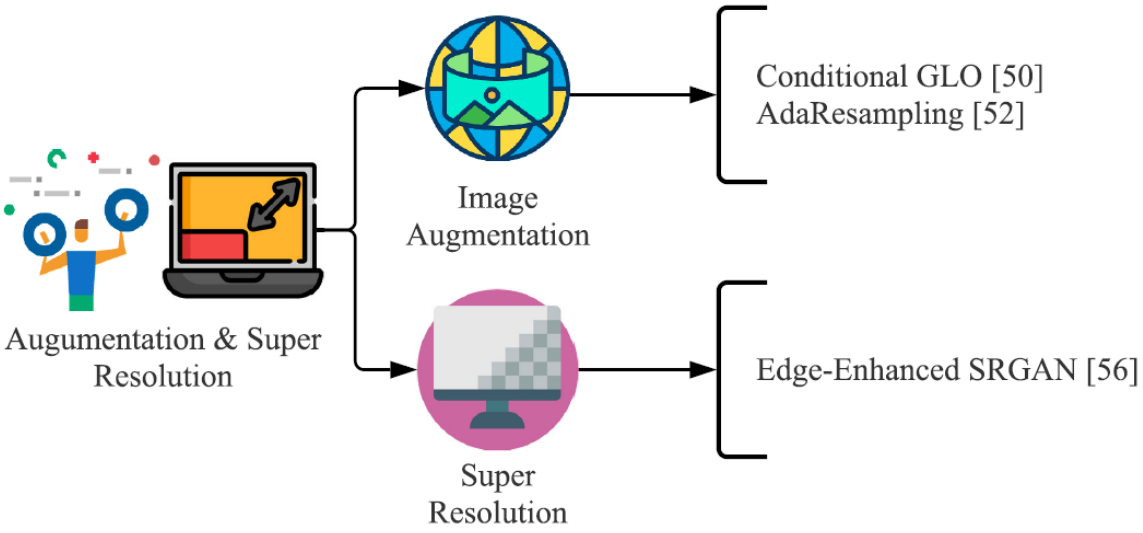
شکل ۱۶ برای افزایش کارایی دو رهیافت عمده وجود دارد: کاهش محاسبات و یا کاهش مجموعه جستجو



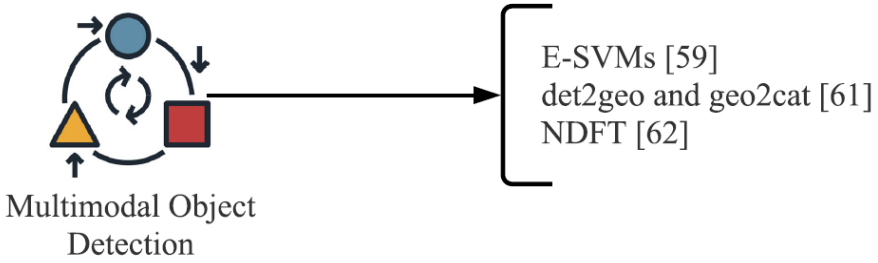
شکل ۱۷ دو روش برای تشخیص اشیا کوچک: تغییر هایپرپارامترهای مدل برای اجسام ریز و یادگیری چند مقیاسه



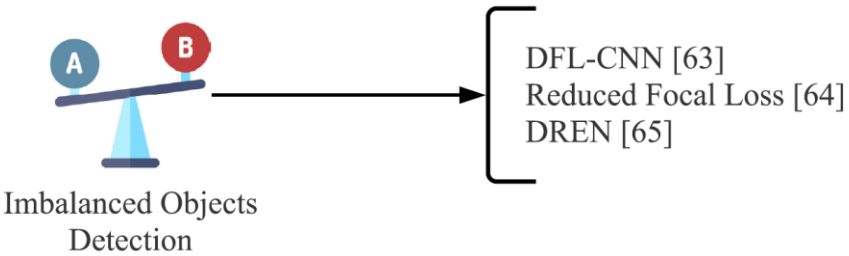
شکل ۱۸ تشخیص صحیح جهت شی با روش هایی مانند rotaion invariant CNN و یا oriented bbox



شکل ۱۹ استفاده از پیش پردازش هایی مانند آگمنتیشن تصاویر و یا افزایش رزولوشن با هدف افزایش دقت مدل



شکل ۲۰ استفاده از ترکیب مدالیته های گوناگون از سنسور های مختلف



شکل ۲۱ حل مشکل عدم تعادل در کلاس ها با روش هایی مانند focal loss و یا difficult region estimation network

الگوریتم های انتخابی

با توجه به پارامترهای مطرح شـده، الگوریتم swin که هم جدید اسـت و هم بنچمارک خوبی از نظر دقت دارد و بر مبنای ترنسفورمر است، برای پیاده سازی انتخاب شده است.

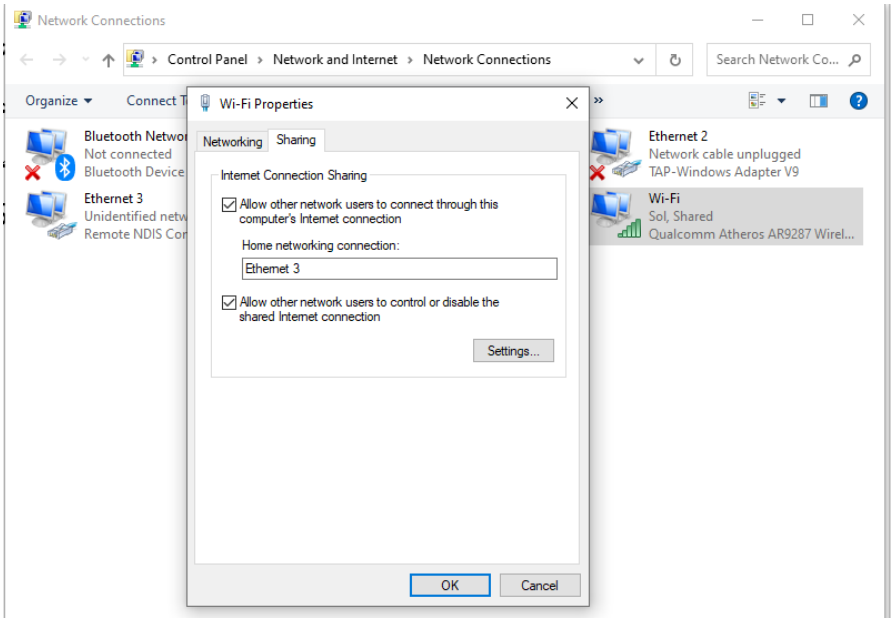
(در این نسخه از گزارش هنوز مرحله ی پیاده سازی مدل انجام نشده است)

پیاده سازی در برد جتسون

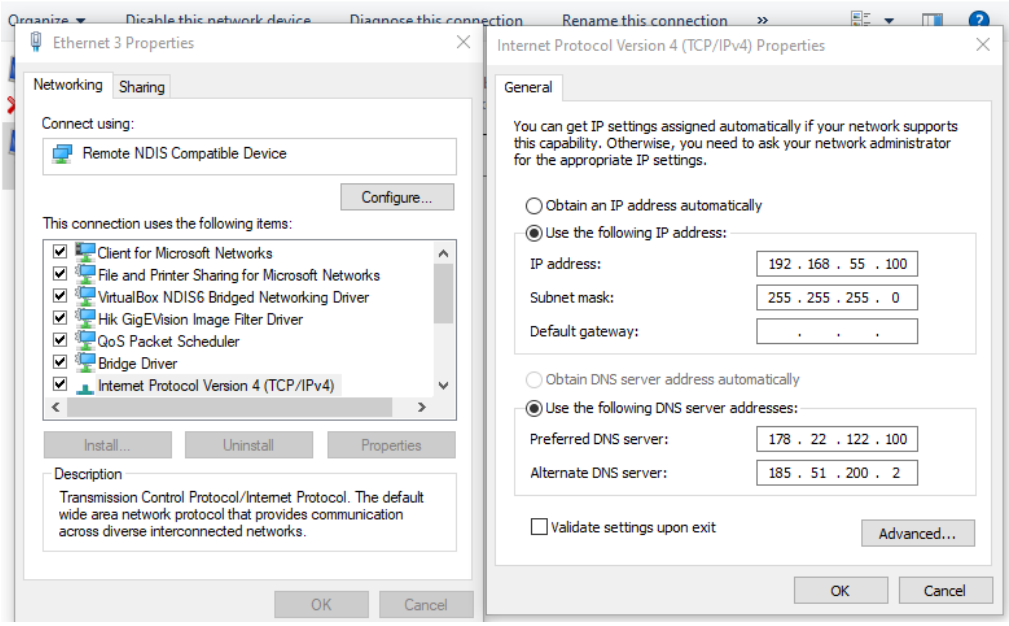
برد جتسون موجود مدل TX2 توسط SDK manager با یک سیستم عامل اوبنتو راه اندازی و JetPack 4.6 بر روی آن نصب شده است.

با اســـتفاده از اتصـــال usb میتوان ارتباط در مد headless بین کامپیوتر و برد ایجاد کرد. در این حالت نیازی به موس، کیبرد و مانیتور اضافه و هم چنین نصب لینوکس (یا ماشین مجازی) نیست. پس از اتصالات لازم (تغذیه micro usb و) و سپس روشن کردن برد بامی توان با ایجاد اتصال ssh به برد از طریق کامپیوتر با سیستم عامل دلخواه دسترسی داشت. ادرس ip برد جتسون 192.168.55.1 می باشد. (دقت شود که طبق داکیومنت ابتدا باید کلیه اتصالات انجام شوند و سپس تغذیه در آخر وصل شود و آن گاه برد روشن شود).

برای اینترنت دهی به برد روش های گوناگونی وجود دارد مانند اتصال کابل شبکه یا استفاده از ماژول وای فای تعبیه شـده و نصـب آنتن ها . اما روش دیگر (که به نظر سـاده تر و کم هزینه تر می آید) اینترنت دهی از همان پورت اتصال usb است. برای این منظور در ویندوز در تنظیمات اینترنت متصل به کامپیوتر (در بخش network connections) مطابق شکل sharing را فعال کرده و اتصال usb را انتخاب میکنیم. ممکن است چندین شبکه مشاهده کنید، بهتر است با مقایسه ی این پوشه قبل و بعد از اتصال برد متوجه نام اتصال شوید. سپس روی اتصال کلید کرده و ip آن را تصحیح کنید (192.168.55.1). برای دیگر سیستم عامل ها روند مشابه قابل اتخاذ است.



شکل ۲۲ فعال سازی اشتراک گذاری اینترنت شبکه ی وایفای در کامپیوتر و انتخاب شبکه ی usb متصل به برد



شکل ۲۳تنظیم IPv4 بر روی ip ، DNS مشاهده شده مربوط به سایت شکن است، البته به نظر می رسد بایستی در etc/resolve.conf اعمال شود.

برای نصب پکیج ها برای رعایت آسانتر ورژن ها و ایزولاسیون بهتر از ایمیج داکری استفاده شد. بر مبنای ریپوی پیشنهاد شده[۲] دستورات زیر برای پول کردن ایمیج l4t-ml حاوی تورچ، تنسورفلو و دیگر کتابخانه های معروف ماشین لرنینگ اجرا شد:

```
git clone https://github.com/dusty-nv/jetson-containers
cd jetson-containers
scripts/docker_run.sh -c nvcr.io/nvidia/l4t-ml:r32.7.1-pth1.7-py3
```

این دستور علاوه بر پول کردن در صورت نبود ایمیج، کانتینر متناظر آن را ساخته و در حالت تعاملی ران می کند. گرچه ورژن L4T جتسون در دسترس 32.7.2 بود ولی از آنجایی که ایمیج داکری برای این ورژن نبود از نسخه ی نزدیک تر استفاده شد. این ایمیج بدون هیچ مشکلی ران شد.

پایتون موجود در این تصویر داکری ورژن 3.6.9 است اما برای شبکه هایی مانند یولو و سوین ترنسفورمر نیاز به پایتون 3.7 می باشد. توجه شود که صرفا با نصب پایتون 3.6.7 مشکل حل نمی شود چرا که کلیه پکیج ها مانند پایتورچ –تا جایی که نگارنده می داند- نیاز به نصب مجدد پیدا می کنند.

[2] https://github.com/dusty-nv

ورژن پنج مدل یولو در داکر هاب دارای ایمیج با معماری arch می باشـد از آنجایی که دارای پایتورچ و پایتون با ورژن مناسب (هم برای یولو هم سوین ترنسفورمر) بود دانلود و اجرا شد. اما پس از تست مشاهده شد که قادر به خواندن GPU برد نمی با شد و روی CPU ران می شود. (البته ایمیج ورژن دو گیگی ت ست شد و جدید تر آن حدود 6GB بود که دیگر تست نشد)

راه حل بعدی که به ذهن رسید استفاده از ایمیج های داکری خود جتسون برای JetPack ورژن های بالاتر(یعنی ورژن ۵ بجای ورژن 4.6 فعلی) بود.چرا که این ایمیج ها دارای پایتون 3.8 می باشند.

تذکر: این گونه ت ست ها به دلیل تحریم ها از آ سو و فیلترها ازین سو گاهی زمانبر ه ستند. مثلا ت ست اخیر که دانلود ایمیج حدود6GB می بود در حین دانلود چندین بار قطع شـد و پس از دانلود نیز موقع اکسـترکت پیغام نبود فضا داد و کل پروسه منتفی شد، سپس برای حذف ایمیج های قبلی و خالی سازی فضا ابتدا با دستور

Sudo docker image save –o filename image: tag

فایل ایمیج را استخراج کرده و با

scp nvidia@192.168.55.1:path/to/filename path/to/local/dir

در محل مورد نظر در ها ست ذخیره می کنیم تا در صورت نیاز بعدا راحت تر با د ستور Load –i ا ستفاده کنیم. (قبل از کپی با chown usr:usr مالکیت فایل را تغییر دهید تا بتوان آن را ترنسـفر کرد، Usr در اینجا nvidia است)

همچنین برای کاهش قطعی دانلود، پیش فرض دانلود موازی سه لایه را به یک لایه تغییر می دهیم:

sudo vim /etc/docker/daemon.json -> add : { "max-concurrent-uploads": 1, "max-concurrent-downloads": 4 }

sudo service docker restart

(در زمان ارسـال این گزارش کار هنوز مرحله ی پیاده سـازی مدل روی برد جتسـون انجام نشـده اسـت. و برای بررسی کارهای انجام شده تا این لحظه برای ناظر ارسال میگردد.)

References

Kang, J. (2022). A Survey of Deep Learning-Based Object Detection Methods and Datasets for Overhead Imagery.

Shao, F. (2021). Deep Learning for Weakly-Supervised Object Detection and Object Localization: A Survey.

Zaidi, S. S. (2021). A Survey of Modern Deep Learning based Object Detection Models.