به نام خدا

یادگیری ماشین آماری

تئوری و توضیحات مرتبط با مدل YOLOv4

استاد: دکتر امینی

پژوهشگر: امیرحسین حیدری

هدف اصلی این داکیومنت ارائه‌ی توضیحی جامع درباره‌ی نسخه‌ی چهارم YOLO (YOLOv4) است. در ابتدا، به مبانی و اصولی که YOLO بر پایه‌ی آن‌ها توسعه یافته است پرداخته می‌شود. سپس، مفاهیمی که در نسخه‌های پیشین YOLO به ساختار مدل و نوع مدل‌سازی اضافه شده‌اند بررسی می‌گردند. در نهایت، در بخش Modern Object Detection، مبانی و اصول YOLOv4 به تفصیل مورد بحث قرار می‌گیرند.

مقدمه

شبکه‌های عصبی کانولوشنی (CNN) نقش مهمی در پیشرفت بینایی کامپیوتری داشته‌اند و مدل‌های متعددی برای تشخیص اشیاء توسعه داده شده‌اند. یکی از روش‌های برجسته در این حوزه مدل YOLO (You Only Look Once) است که رویکردی تک‌مرحله‌ای برای تشخیص اشیاء ارائه می‌دهد. این مدل در مقایسه با روش‌های دومرحله‌ای مانند R-CNN و Faster R-CNN سرعت بیشتری دارد و دقت بالایی ارائه می‌دهد.

YOLO توانسته است تحول بزرگی در بینایی کامپیوتری ایجاد کند و از این مدل در کاربردهای متعددی مانند نظارت تصویری، اتومبیل‌های خودران، تشخیص چهره، تحلیل تصاویر پزشکی، و حتی صنایع هوشمند استفاده می‌شود. این روش به دلیل کارایی بالا و سرعت بالا در محیط‌های عملی و زمان واقعی، توجه محققان و مهندسان بسیاری را به خود جلب کرده است.

تشخیص اشیاء: رویکرد تک‌مرحله‌ای در برابر دومرحله‌ای

در حوزه‌ی تشخیص اشیاء، دو رویکرد اصلی وجود دارد:

روش‌های دومرحله‌ای (Two-Stage) مانند R-CNN و Faster R-CNN که ابتدا مناطق پیشنهادی (Region Proposals) را تولید کرده و سپس این مناطق را برای تشخیص اشیاء طبقه‌بندی می‌کنند.

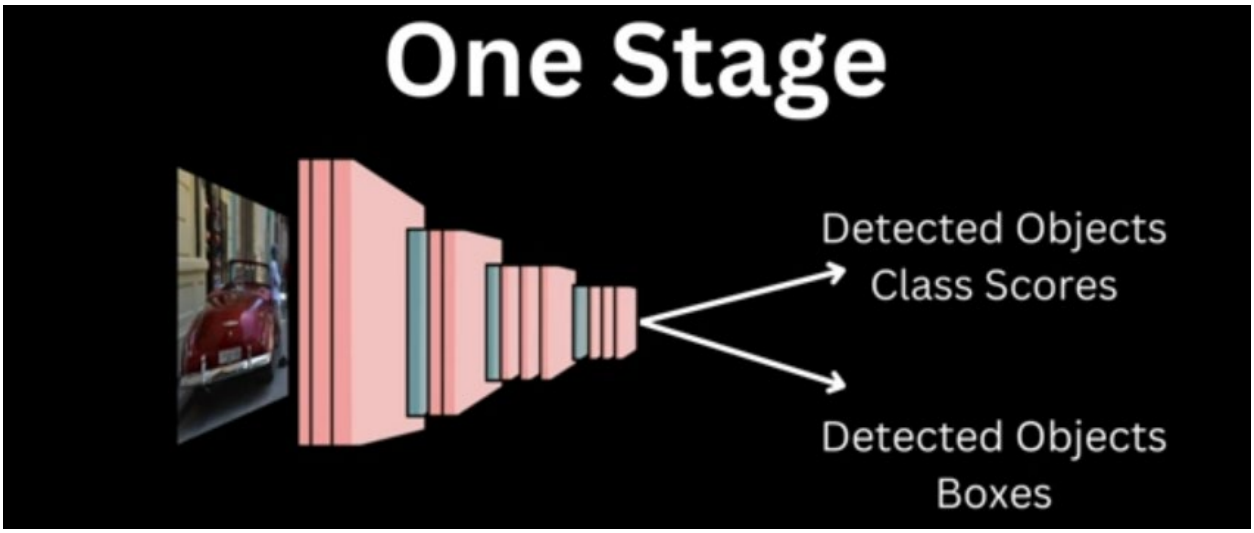
روش‌های تک‌مرحله‌ای (One-Stage) مانند YOLO و SSD که کل فرآیند تشخیص را در یک مرحله انجام می‌دهند.

مزایای هر روش:

روش‌های دومرحله‌ای دقت بالاتری دارند، اما سرعت پایین‌تری دارند زیرا ابتدا باید مناطق پیشنهادی را استخراج کرده و سپس آنها را پردازش کنند.

روش‌های تک‌مرحله‌ای سریع‌تر هستند، زیرا کل فرآیند را در یک شبکه‌ی واحد انجام می‌دهند. اما این روش‌ها معمولاً در تشخیص اشیای کوچک یا متراکم عملکرد ضعیف‌تری دارند.

مدل YOLO یک روش تک‌مرحله‌ای است که از ابتدا برای سرعت بالا طراحی شده است. این روش برخلاف مدل‌های مبتنی بر Region Proposal، تصویر را به صورت یک شبکه‌ی واحد تقسیم کرده و مکان اشیاء را مستقیماً پیش‌بینی می‌کند



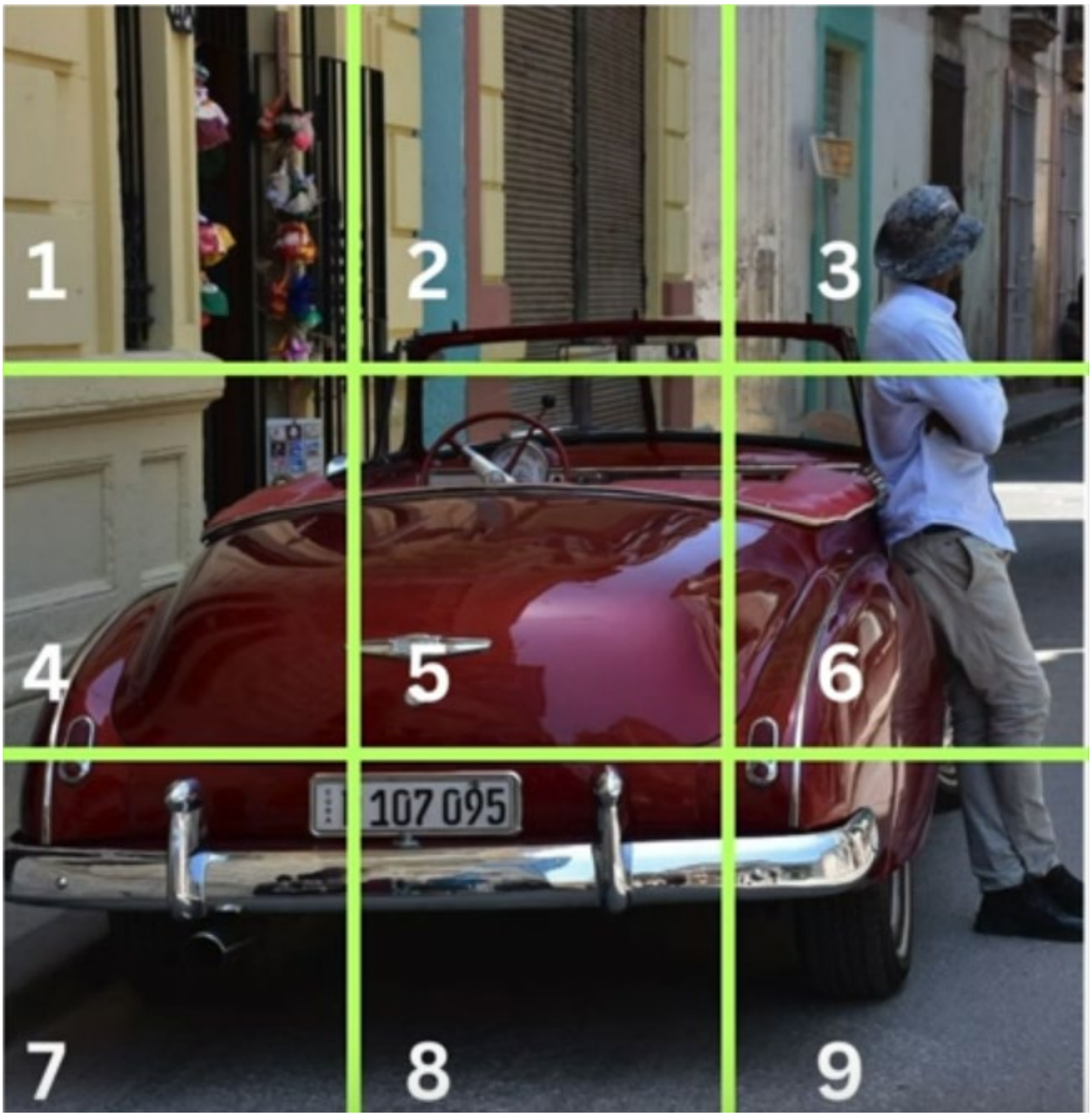
YOLO: ایده‌ی اولیه و پیشرفت‌ها.

YOLOدر ابتدا توسط Joseph Redmon معرفی شد. این روش برخلاف مدل‌های مبتنی برRegion Proposal ، تصویر را به صورت یک شبکه‌ی واحد تقسیم کرده و مکان اشیاء را مستقیماً پیش‌بینی می‌کند YOLO .در نسخه‌های مختلف خود پیشرفت‌های چشمگیری داشته است:

YOLOv1 (2016)

اولین نسخه‌ی YOLO با هدف دستیابی به سرعت بالا طراحی شد و به صورت زیر عمل می‌کرد:

تصویر ورودی را به یک شبکه‌ی S×S تقسیم می‌کرد.



در این تصویر تقسیم بندی ۳ در ۳ بوده است.

هر سلول مسئول تشخیص اشیایی بود که مرکز آنها در آن قرار دارد.

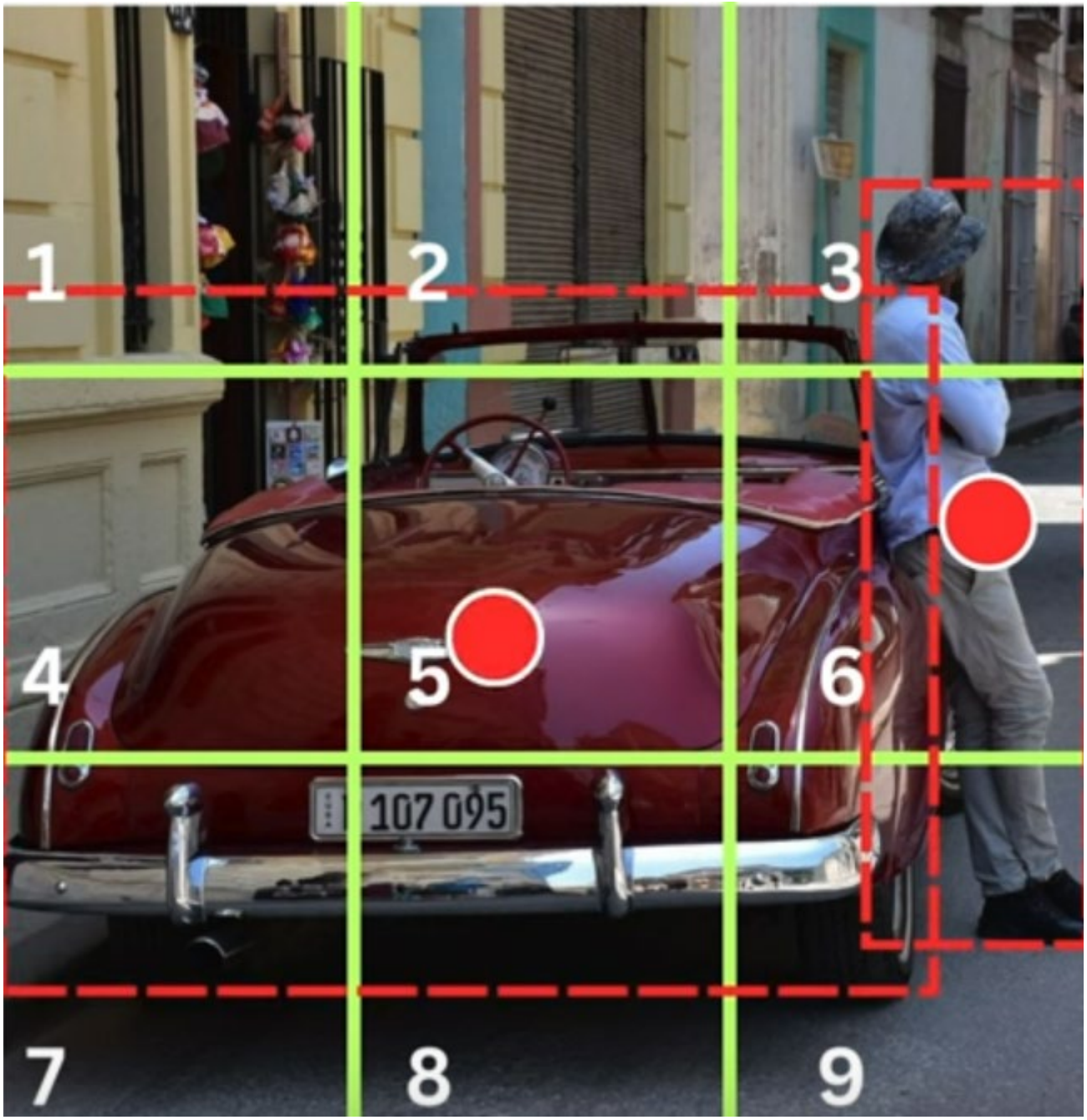
از Bounding Box Regression برای پیش‌بینی مختصات استفاده می‌شد.

سرعت بالایی داشت اما در تشخیص اشیای کوچک یا متراکم عملکرد ضعیفی داشت.

این مدل مزایای زیادی داشت، اما با چالش‌هایی مانند عدم دقت در تشخیص اشیای کوچک، تداخل در اشیای متراکم و ناتوانی در یادگیری جزئیات پیچیده مواجه بود. با این حال، YOLOv1 پایه‌ی محکمی برای توسعه‌ی نسخه‌های بعدی شد.

۳.۱ Bounding Box Regression

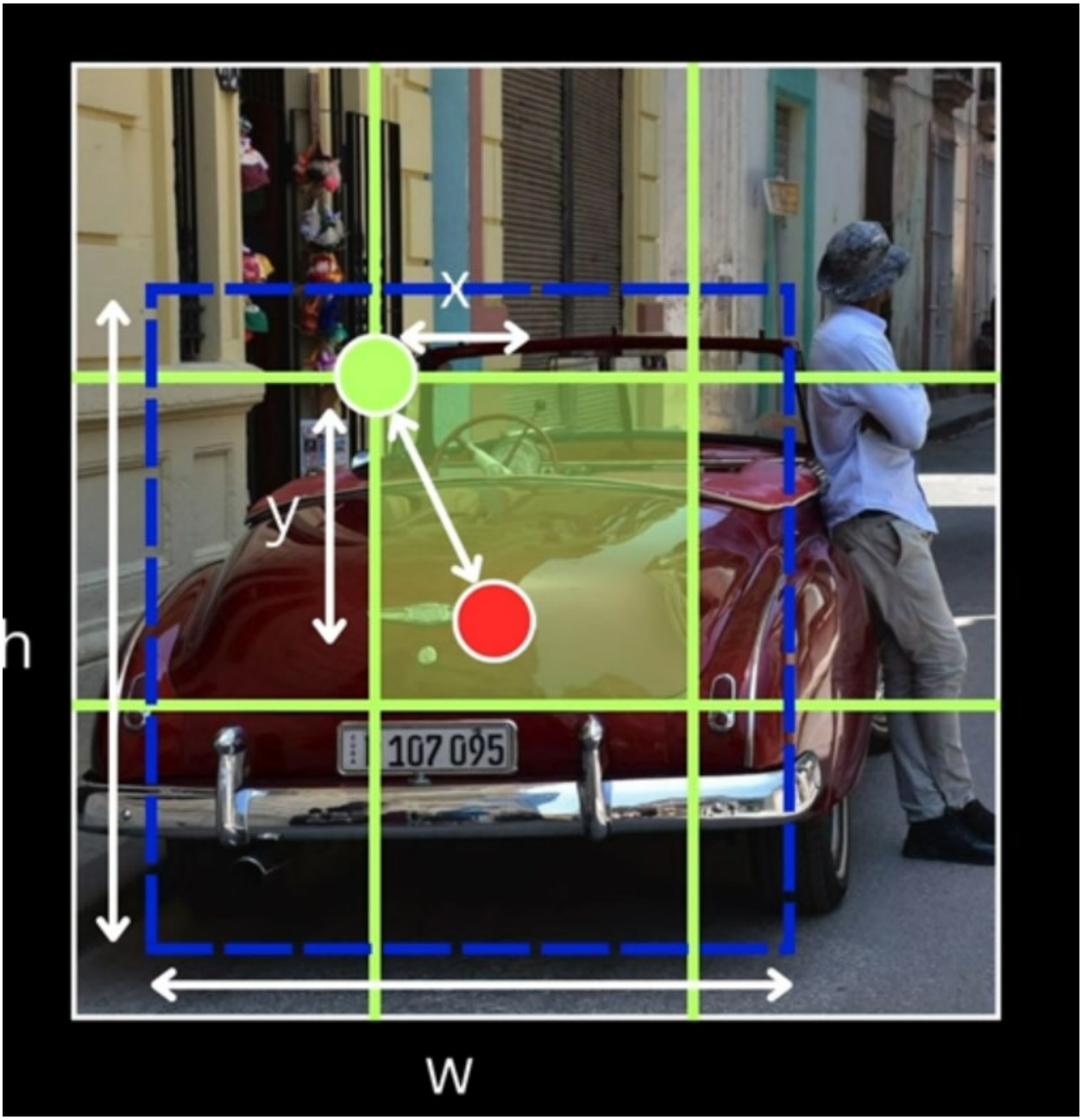
۳. رگرسیون جعبه محدودکننده(Bounding Box Regression)



یکی از اجزای کلیدی در مدل YOLO رگرسیون جعبه محدودکننده (Bounding Box Regression) است. این فرآیند مسئول تعیین مختصات دقیق جعبه‌ای است که شیء را در تصویر محصور می‌کند .YOLO تصویر را به یک شبکه‌ی S×S تقسیم کرده و هر سلول وظیفه‌ی پیش‌بینی جعبه‌های محدودکننده را برعهده دارد. هر جعبه با استفاده از چهار پارامتر زیر توصیف می‌شود:

- مختصات مرکز جعبه نسبت به محدوده‌ی سلول شبکه.
- عرض و ارتفاع جعبه نسبت به ابعاد کل تصویر.

در تصویر زیر، فرآیند پیش‌بینی جعبه‌های محدودکننده نشان داده شده است:



تصویر به یک شبکه‌ی ۳×۳ تقسیم شده است (خطوط سبز).

دو شیء اصلی شامل یک خودرو قرمز و یک فرد ایستاده در سمت راست تصویر شناسایی شده‌اند نقاط قرمز در مرکز هر شیء مشخص می‌کنند که کدام سلول مسئول تشخیص آن شیء است. جعبه‌های محدودکننده (خطوط قرمز نقطه‌چین) نشان‌دهنده‌ی محدوده‌ی تخمینی هر شیء هستند.

مختصات جعبه برای خودرو قرمز:

مرکز خودرو در سلول ۵ قرار دارد، یعنی سلول وسطی شبکه. فرض کنید مرکز خودرو در مختصات نسبی سلول، $x=0.4$ و $y=0.5$ باشد.

اندازه‌ی جعبه (Bounding Box) را فرض کنیم:

عرض خودرو $w=0.6$ نسبت به تصویر کامل (یعنی $0.6 \times 900 = 540$ پیکسل).

ارتفاع خودرو $h=0.3$ نسبت به تصویر کامل (یعنی $0.3 \times 900 = 270$ پیکسل).

بنابراین، مختصات نهایی جعبه محدودکننده برای خودرو در تصویر کامل چنین خواهد بود:

مرکز: $(300+0.4\times300, 300+0.5\times300) = (420, 450)$

عرض: ۵۴۰ پیکسل

ارتفاع: ۲۷۰ پیکسل

این متغییرها در Yolo مورد استفاده قرار میگیرند.

YOLOچگونه مختصات جعبه‌ها را پیش‌بینی می‌کند؟

برای هر سلول در شبکه، YOLOچهار مقدار کلیدی را پیش‌بینی می‌کند:

موقعیت مرکز شیء نسبت به محدوده‌ی سلول.

عرض و ارتفاع جعبه نسبت به کل تصویر.

این پیش‌بینی‌ها با استفاده از تابع هزینه‌ی CIoU (Complete IoU Loss) اصلاح می‌شوند تا دقت در قرارگیری جعبه‌ها افزایش یابد.

در تصویر، خودرو در سلول شماره ۵ قرار دارد، بنابراین مختصات جعبه‌ی محدودکننده‌ی آن از این سلول محاسبه می‌شود. فرد ایستاده در سمت راست در سلول ۶ و ۹ قرار دارد و Bounding Box او نیز از این سلول‌ها به دست می‌آید.

YOLOv2 (2017) - YOLO9000

YOLOv2 به عنوان نسخه‌ای بهبودیافته از YOLOv1 معرفی شد و چندین ویژگی کلیدی را ارائه داد:

استفاده از Anchor Boxes برای بهبود دقت مکان‌یابی اشیاء Anchor Boxes تکنیکی است که به مدل کمک می‌کند تا اشیاء را در تصویر دقیق‌تر شناسایی کند. تصور کنید که مدل باید یک جعبه‌ی دور هر شیء بکشد تا آن را مشخص کند. اگر مدل بخواهد این جعبه را از ابتدا بکشد، احتمال زیادی دارد که اندازه و موقعیت دقیق آن را اشتباه تخمین بزند. اما اگر چندین جعبه‌ی آماده با ابعاد و شکل‌های مختلف در تصویر داشته باشیم، مدل فقط یاد می‌گیرد که کدام جعبه به کدام شیء نزدیک‌تر است و آن را کمی جابه‌جا کند تا به محل دقیق برسد. این روش باعث می‌شود مدل دقت بالاتری داشته باشد و مکان‌یابی بهتری انجام دهد.

معماری بهینه‌تر و استفاده از Batch Normalization که باعث افزایش پایداری آموزش شد Batch Normalization روشی است که مدل را در طول یادگیری پایدارتر و سریع‌تر می‌کند. فرض کنید مدل باید از روی تصاویر یاد بگیرد که چه اشیائی در آن‌ها وجود دارد. اما اگر روشنایی تصاویر متفاوت باشد یا شدت رنگ‌ها تغییر کند، مدل ممکن است در یادگیری دچار مشکل شود . Batch Normalizationداده‌های ورودی را طوری تنظیم می‌کند که همه‌ی آن‌ها در یک محدوده‌ی مناسب قرار بگیرند. این کار باعث می‌شود که شبکه‌ی عصبی یادگیری یکنواخت‌تری داشته باشد و در نتیجه دقت و سرعت یادگیری افزایش پیدا کند.

بهبود آموزش چندکلاسه‌ای و افزایش توانایی در تشخیص ۹۰۰۰ دسته از اشیاء با استفاده از یادگیری چندسطحی. یادگیری چندسطحی (Hierarchical Learning) به معنای استفاده از اطلاعات در سطوح مختلفی از جزئیات است. در این روش، مدل ابتدا ویژگی‌های کلی و عمومی یک شیء را یاد می‌گیرد (مثلاً تشخیص اینکه یک شیء یک حیوان است)، سپس در سطوح بعدی، جزئیات بیشتری مانند نوع حیوان (مثلاً سگ یا گربه) را بررسی می‌کند و در نهایت، حتی نژاد خاص آن را شناسایی می‌کند (مثلاً سگ از نژاد هاسکی یا لابرادور). در YOLOv2، این روش باعث شد که مدل بتواند بین دسته‌بندی‌های عمومی و جزئیات دقیق تفاوت قائل شود و در نتیجه دقت بیشتری در تشخیص و دسته‌بندی اشیاء داشته باشداستفاده از Fine-Grained Features برای استخراج ویژگی‌های دقیق‌تر Fine-Grained Featuresبه این معناست که مدل می‌تواند تفاوت‌های جزئی بین اشیای مشابه را تشخیص دهد. فرض کنید می‌خواهیم مدل بتواند بین انواع مختلف پرنده یا مدل‌های مختلف ماشین تمایز قائل شود. برای این کار، مدل باید بتواند ویژگی‌های دقیق‌تری از اشیاء را یاد بگیرد، مانند شکل خاص بال‌های یک پرنده یا طراحی چراغ‌های یک مدل خاص از خودرو. درYOLOv2 ، لایه‌های شبکه‌ی عصبی طوری طراحی شده‌اند که بتوانند این جزئیات را بهتر استخراج کنند. در نتیجه، مدل دقت بیشتری در دسته‌بندی اشیاء دارد و می‌تواند حتی تفاوت‌های بسیار کوچک را نیز تشخیص دهد.

با این تغییرات، YOLOv2توانست به دقت بالاتری برسد و کاربردهای گسترده‌تری پیدا کند. اما همچنان در شناسایی اشیای کوچک و متراکم با مشکلاتی مواجه بود.

YOLOv3 (2018)تشخیص اشیاء با دقت و سرعت بیشتر

در سال ۲۰۱۸، نسخه سوم YOLO یعنی YOLOv3 معرفی شد که چندین بهبود کلیدی در مقایسه با نسخه‌های قبلی خود داشت. این نسخه به دلیل افزایش دقت و توانایی تشخیص اشیاء در اندازه‌های مختلف مورد توجه قرار گرفت.

بهبودهای کلیدی درYOLOv3

معماری جدید باDarknet-53

درYOLOv3 ، معماری شبکه عصبی ازDarknet-19 بهDarknet-53 ارتقا یافت. این تغییر باعث افزایش قابلیت استخراج ویژگی‌های تصویری شد و دقت مدل را بهبود بخشید.

در سیستم‌های نظارتی، Darknet-53می‌تواند چهره افراد را با دقت بیشتری شناسایی کند، حتی اگر نور کم یا زاویه تصویر نامناسب باشد.

۲ .تشخیص اشیاء در سه مقیاس مختلف

YOLOv3می‌تواند اشیاء را در سه اندازه مختلف شناسایی کند:

-اشیاء کوچک (مانند پرندگان یا نوشته‌های تابلوها)

-اشیاء متوسط

-اشیاء بزرگ (مانند خودروها و ساختمان‌ها)

گر یک تصویر شامل هم یک ماشین و هم یک گربه کوچک باشد، YOLOv3 می‌تواند هر دو را به‌طور هم‌زمان و دقیق شناسایی کند.

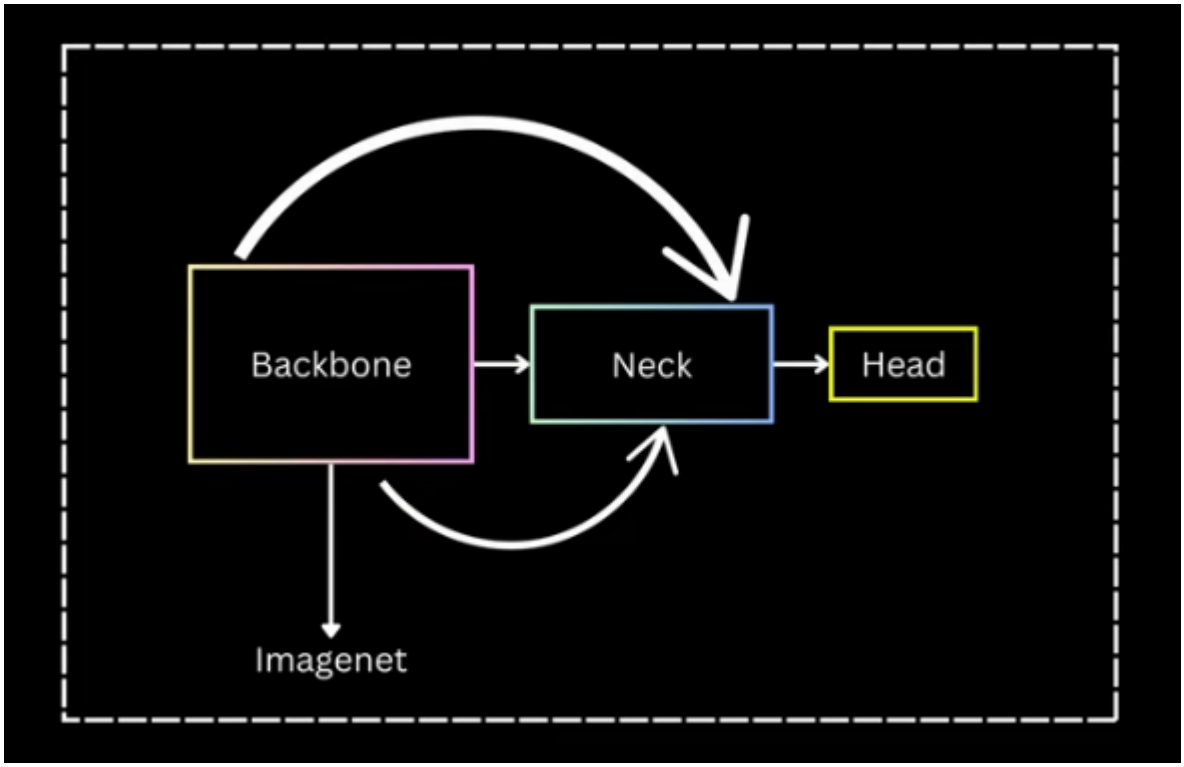
بهینه‌سازی محاسباتی با CSPNet

این نسخه با استفاده از تکنیک Cross-Stage Partial Networks (CSPNet) توانست میزان محاسبات را کاهش دهد بدون اینکه دقت مدل افت کند. مانند خواندن یک متن طولانی در بخش‌های کوچک است که پردازش آن را سریع‌تر و کارآمدتر می‌کند.

YOLOv3 یکی از مهم‌ترین نسخه‌های YOLO است که تعادلی بین دقت و سرعت برقرار کرده و در بسیاری از کاربردهای عملی، انتخابی مناسب محسوب می‌شود.

Modern object detector

به‌طور معمول، یک Modern object detector دو بخش دارد: بخش ستون فقرات (backbone) و بخش سر (head) ستون فقرات روی دیتاست ImageNet پیش‌تمرین می‌شود و وظیفه‌ی استخراج ویژگی‌ها را بر عهده دارد، در حالی که بخش سر مسئول پیش‌بینی کلاس‌ها و جعبه‌های محدودکننده (bounding boxes) است. برخی معماری‌های رایج برای ستون فقرات در پلتفرم‌های GPU شامل VGG ، ResNetو DenseNet هستند و در پلتفرم‌های CPU نیز از SqueezeNet ، MobileNet و ShuffleNet استفاده می‌شود.

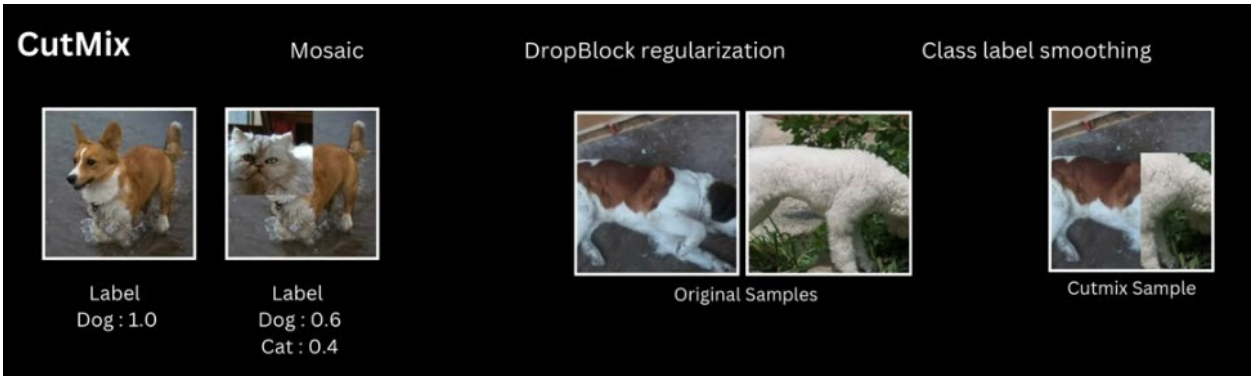


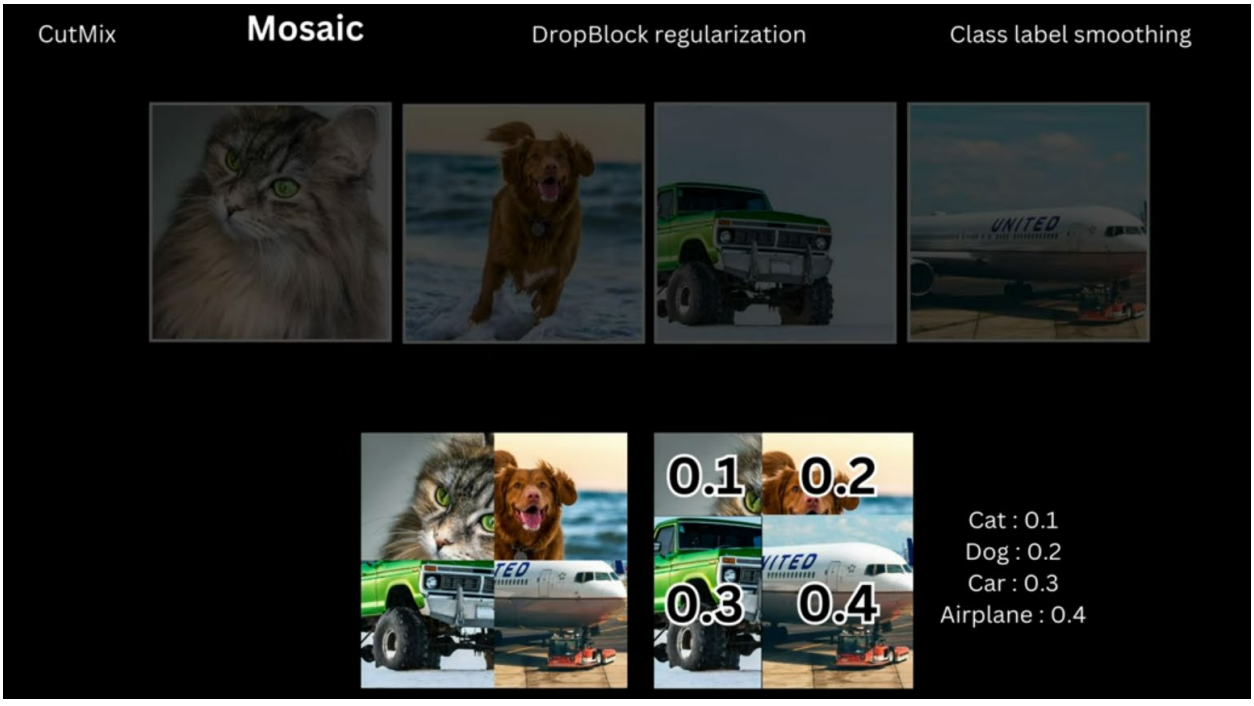
به‌طور کلی، Modern object detector بر اساس ساختار «سر» به دو دسته‌ی تک‌مرحله‌ای (one-stage) و دومرحله‌ای (two-stage) تقسیم می‌شوند. مجموعه R-CNN ، از جمله Faster R-CNN ، R-FCN وLibra R-CNN ، نمونه‌های روش‌های دومرحله‌ای به‌شمار می‌روند، در حالی کهYOLO ، SSDو RetinaNet مثال‌هایی از روش‌های تک‌مرحله‌ای هستند.

امروزه، برای ساخت Modern object detector ، بین ستون فقرات و سر، لایه‌هایی قرار داده می‌شود که **گردن (neck)** نام دارند و feature maps را از سطوح مختلف جمع‌آوری و تلفیق می‌کنند. شبکه‌های مختلفی از جملهFPN ، PANet، BiFPN و NAS-FPN را می‌توان به‌عنوان بخش گردن در یک شبکه به‌کار برد. پیش از معرفی معماری اصلیYOLOv4 ، چند مفهوم کلیدی در ادامه بیان شده است.

Bag of freebies

به مجموعه‌ای از روش‌های آموزشی گفته می‌شود که می‌توانند بدون افزایش هزینه‌ی استنتاج، دقت object detector را بهبود دهند. تنها تغییری که اتفاق می‌افتد در روند آموزش (یا مدت‌زمانِ آن) است. در مبحث تشخیص اشیاء،Bag of freebies معمولاً به داده‌افزایی (data augmentation) اشاره دارد؛ داده‌افزایی باعث می‌شود تنوع تصاویر ورودی بیشتر شود و مدل در مواجهه با محیط‌ها و شرایط نوری گوناگون مقاوم‌تر عمل کند. دو گروه متداول از روش‌های داده‌افزایی عبارت‌اند از photometric distortions و geometric distortions در photometric distortions ، مؤلفه‌هایی نظیر روشنایی، کنتراست، ته‌رنگ(hue) ، اشباع (saturation) و نویز دستکاری می‌شوند و در geometric distortions از تغییر مقیاس تصادفی، برش(crop) ، وارون‌سازی (flip) و چرخش (rotate) استفاده می‌گردد.





Bag of specials

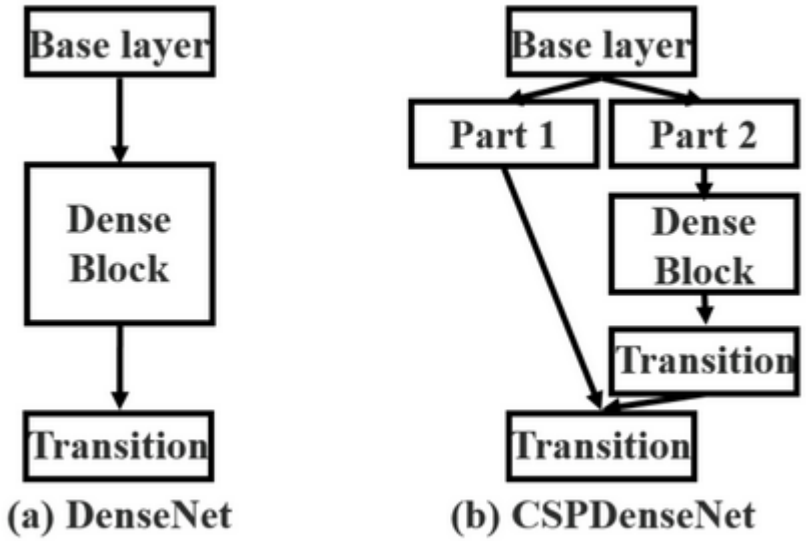
Bag of specials شامل بعضی ماژول‌های جانبی (plugin) و روش‌های پس‌پردازش (post-processing) است که اندکی هزینه استنتاج را افزایش می‌دهند اما دقت object detector را در مقیاس وسیع ارتقاء می‌بخشند. این ماژول‌ها می‌توانند میدان دید (receptive field) را گسترش داده، مکانیزم‌های توجه **(attention)** را معرفی کرده یا قابلیت‌های ادغام ویژگی‌ها را تقویت کنند و ....، همچنین، فرایند پس‌پردازش برای ارزیابی و گزینش نتایج نهایی مدل به‌کار می‌رود. از میان ماژول‌های رایج در این دسته می‌توان به SPP، ASPP و RFB اشاره کرد.

در معماری CSPDenseNet ، ابتدا ویژگی‌های خروجی از لایه‌ی پایه (base layer) به دو قسمت تقسیم می‌شوند. سپس این دو بخش در ساختاری به نام میان‌مرحله‌ای (cross-stage) دوباره با هم ترکیب می‌شوند تا ضمن حفظ یا حتی کاهش هزینه‌ی محاسباتی، عملکرد شبکه بهبود یابد.

دو بخش اصلی در این معماری عبارت‌اند از:

بلوک چگال جزئی(partial dense block)

لایه‌ی گذار جزئی(partial transition layer)



partial dense block مسیر عبور گرادیان را طولانی‌تر می‌کند؛ به این ترتیب فشار محاسباتی بین لایه‌های شبکه تقسیم می‌شود و حجم داده‌های در حال عبور (ترافیک حافظه) کاهش می‌یابد. در لایه‌ی partial dense block نیز از روشی به نام «کوتاه‌سازی جریان گرادیان» (truncating the gradient flow) استفاده می‌شود تا لایه‌ها گرادیان را به شکل تکراری یاد

نگیرند و ترکیب گرادیان‌های به دست آمده بیشترین تنوع ممکن را داشته باشد. این فرایند در نهایت باعث بالا رفتن کارایی کل شبکه می‌شود.

در یک بلوک partial dense، feature map لایه پایه به دو بخش تقسیم می‌شوند. $(X_0 = [X_0', X_0''])$ X_0' مستقیماً به انتهای مرحله متصل می‌شود و X_0'' از طریق یک بلوک dense عبور می‌کند. خروجی لایه‌های dense $[X_0', X_1, \dots, X_k]$

وارد یک لایه انتقال می‌شود. سپس خروجی این لایه انتقال (X_T) با X_0'' ترکیب می‌شود و از یک لایه انتقال دیگر عبور می‌کند. در نهایت، خروجی X_U تولید می‌شود. معادلات مسیر پیش‌رو در معادلات زیر نشان داده شده‌است.

$$X_k = W_k * [X_0', X_1, \dots, X_{k-1}]$$

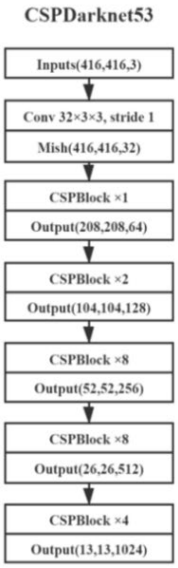
$$X_T = W_T * [X_0', X_1, \dots, X_k]$$

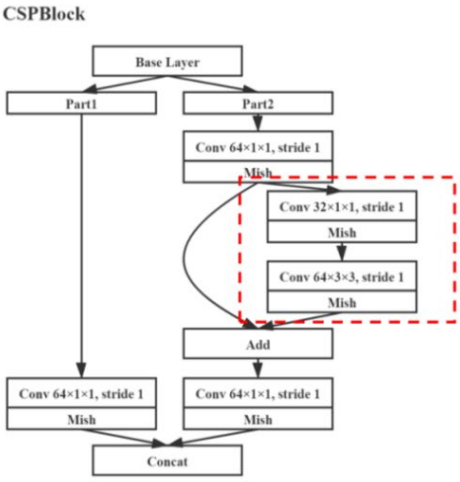
$$X_U = W_U * [X_0'', X_T]$$

CSPDarknet53

CSPDarknet53 ترکیبی از دو شبکه Darknet53 و CSPDenseNet است. این شبکه یک شبکه عصبی کانولوشنی است که به عنوان پایه‌ای برای تشخیص اشیاء عمل می‌کند. شبکه Darknet53 شامل 53 لایه کانولوشنی است. این 53 لایه کانولوشنی شامل لایه‌هایی با اندازه‌های 1 × 1 و 3 × 3 است. از این میان، 29 لایه کانولوشنی با ابعاد 3 × 3 هستند. هر لایه کانولوشن به یک لایه نرمال‌سازی دسته‌ای (Batch Normalization یا BN) و یک لایه فعال‌سازی Mish متصل است. معماری‌های شبکه‌های Darknet53 و CSPDarknet53 در شکل زیر آورده شده است. در مقاله‌ی اصلی، از CSPDarknet53 به عنوان بخش اصلی (backbone) برای YOLOv4 استفاده شده است، در حالی که Darknet53 به عنوان بخش اصلی برای YOLOv3 به کار رفته است.

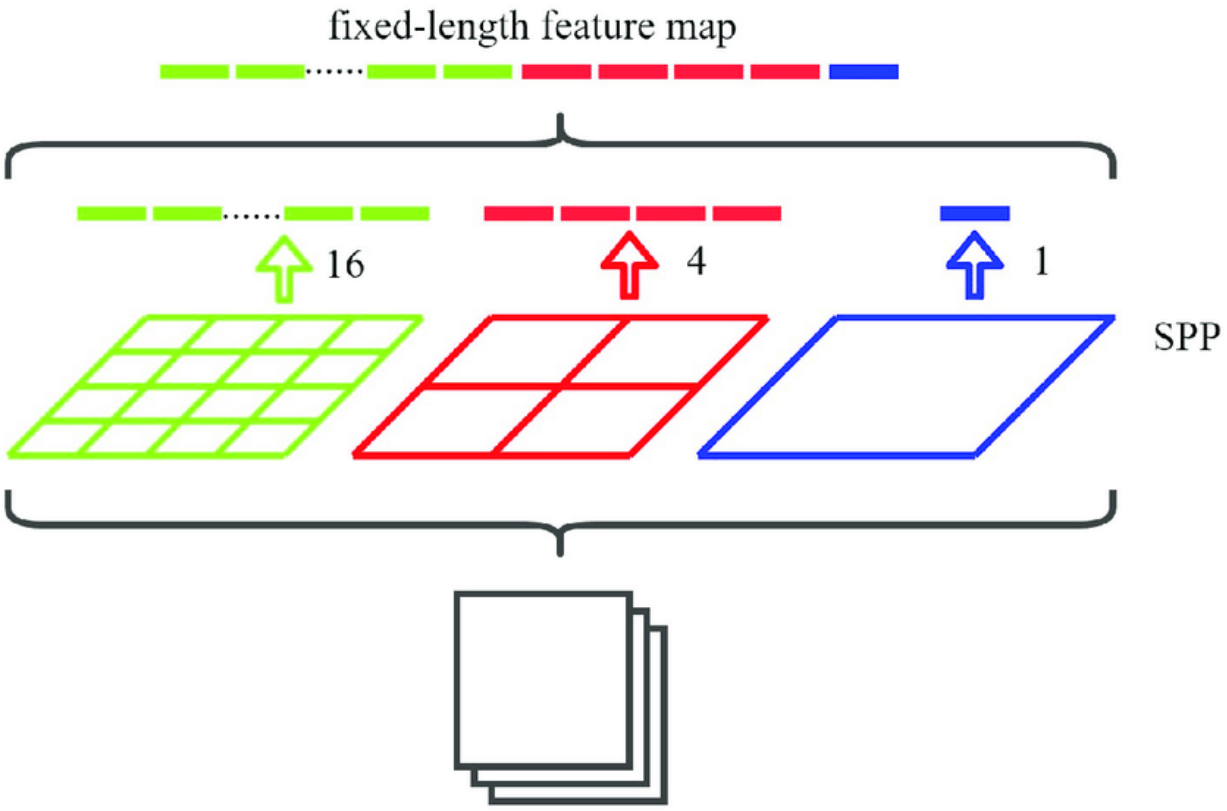
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8





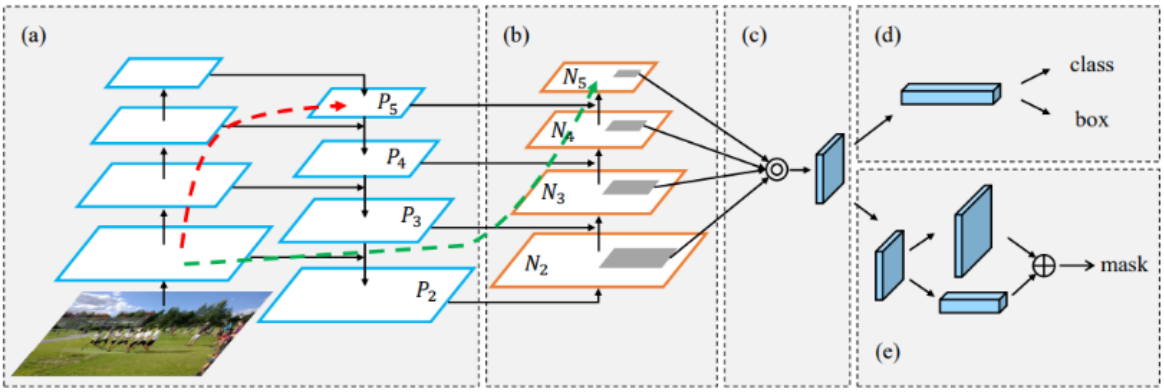
Spatial Pyramid Pooling

Spatial Pyramid Pooling (SPP) به شبکه کمک می‌کند تا از اندازه‌های مختلف تصاویر استفاده کند، برخلاف سیستم‌های سنتی که نیاز به تصویر ورودی با اندازه ثابت دارند. این روش از لایه‌های مکس‌پولینگ با اندازه‌های مختلف ۵ ، ۹ و ۱۳ برای استخراج ویژگی‌ها استفاده می‌کند و خروجی‌هایی با اندازه ثابت تولید می‌کند. به عبارت دیگر، Spatial Pyramid Pooling به شبکه این امکان را می‌دهد که ویژگی‌های مختلف را از ورودی‌هایی با اندازه‌های متفاوت استخراج کرده و در نهایت خروجی‌ای با اندازه ثابت تولید کند. درYOLOv4 ، از یک بلوک SPP بعد از CSPDarknet53 برای استخراج ویژگی‌های مهم از بخش اصلی (backbone) استفاده می‌شود.



Path Aggregation Network

Path Aggregation Network (PANet)، که یک feature pyramid networ است، در YOLOv4 برای تقویت اطلاعات مربوط به بخش segmentation به کار می‌رود. این شبکه از رویکردهای بالا به پایین و پایین به بالا استفاده می‌کند تا ویژگی‌های مهم را از مقیاس‌های مختلف طبقه‌بند اصلی (backbone) استخراج کرده و عملکرد مدل را بهبود بخشد. خروجی PANet (detection heads) ارسال می‌شود. در شکل زیر {P2، P3، P4، P5} برای نشان دادن سطوح ویژگی‌های تولید شده توسط بخش اصلی (backbone) استفاده می‌شود. این فرآیند از P2 شروع می‌شود و به P5 می‌رسد، و در این مسیر، ویژگی‌ها به تدریج کاهش می‌یابند. {N2، N3، N4، N5} نقشه‌های ویژگی جدیدی هستند که تولید می‌شوند. هر یک از این سطوح از یک لایه کانولوشن ۳ × ۳ عبور می‌کند.



(a) بخش اصلی (Backbone) شبکه ویژگی‌های هرمی.(FPN)

(b) تقویت مسیر پایین به بالا.(Bottom-up path augmentation)

(c) تجمیع ویژگی‌های تطبیقی.(Adaptive feature pooling)

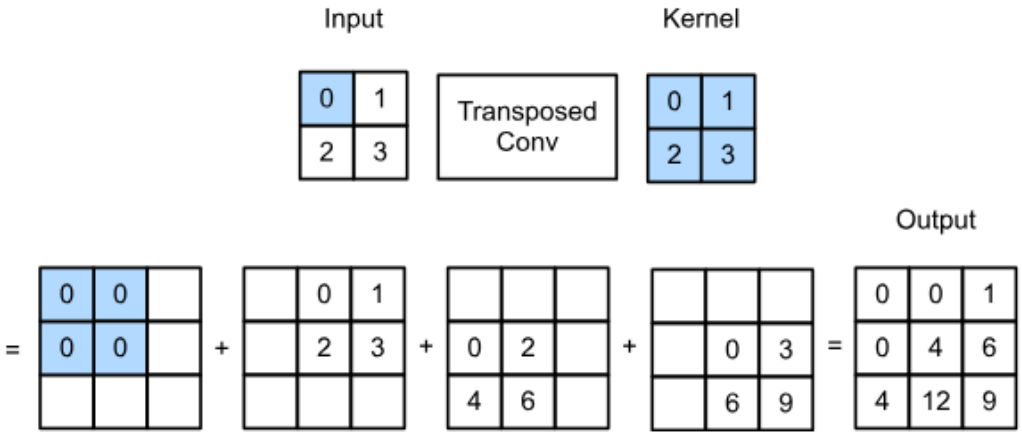
(d) شاخه جعبه.(Box branch)

(e) ترکیب کاملاً متصل (Fully-connected fusion)

مرحله فعال‌سازی و تجمیع ویژگی‌های تطبیقی

در این مرحله، پس از یک لایه کانولوشنی(Convolutional Layer) ، یک لایه فعال‌سازی ReLU اعمال می‌شود. در مرحله تجمیع تطبیقی ویژگی‌ها(Adaptive Feature Pooling) ، از Region of Interest Align (RoIAlign) برای عملیات تجمیع ویژگی‌ها استفاده می‌شود. این روش امکان استخراج اطلاعات پارامتری از سطوح بالاتر را فراهم می‌کند. در ادامه، اطلاعات حاصل از لایه‌های مختلف توسط عملیات (Element-wise Max Operation)با یکدیگر ادغام می‌شوند تا ویژگی‌های مهم‌تر تقویت شوند.

در مرحله اتصال کامل(Fully-Connected Stage) ، چهار لایه کانولوشنی متوالی (Consecutive Convolutional Layers)و یک لایه کانولوشن ترانهاده (Transposed Convolutional Layer) به کار گرفته می‌شود. در این مرحله، ویژگی‌ها با استفاده از لایه کانولوشن ترانهاده مورد افزایش مقیاس (Up-Sampling) قرار می‌گیرند. عملیات کانولوشن ترانهاده (Transposed Convolution Operation)معکوس فرآیند کانولوشن استاندارد عمل می‌کند و از پدینگ (Padding) برای افزایش ابعاد استفاده می‌کند.



پیش‌بینی ماسک دودویی(Binary Pixel-Wise Mask)

در این فرآیند، یک ماسک دودویی (Binary Mask) به‌صورت مستقل برای هر کلاس پیش‌بینی می‌شود تا وظایف قطعه‌بندی (Segmentation)و طبقه‌بندی (Classification) از یکدیگر تفکیک شوند. این روش باعث افزایش دقت در شناسایی و تفکیک اشیاء در تصویر می‌شود.

یک ماسک دودویی، تصویری با همان ابعاد تصویر اصلی است که در آن پیکسل‌هایی که به یک شیء خاص تعلق دارند با مقدار ۱ (سفید) و سایر پیکسل‌ها با مقدار ۰ (سیاه) مشخص می‌شوند. این کار باعث می‌شود که مدل بتواند نواحی موردنظر را از پس‌زمینه جدا کند.

سربخش مدل(Head)

سیستم چهار ویژگی (t_x, t_y, t_w, t_h) را برای جعبه‌های محدودکننده پیش‌بینی می‌کند. اگر (c_x, c_y) مختصات گوشه بالا-چپ جعبه محدودکننده باشد و (w_{gt}, h_{gt}) و(w, h)به ترتیب ارتفاع و عرض جعبه محدودکننده واقعی و پیش‌بینی‌شده باشند، در این صورت پیش‌بینی‌ها به‌صورت زیر خواهند بود:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$w = w_{gt} e^{t_w}$$

$$h = h_{gt} e^{t_h}$$

در اینجا، (b_x, b_y)مختصات مرکز جعبه پیش‌بینی‌شده را نشان می‌دهد و $\sigma(t_x)$ و $\sigma(t_y)$ به ترتیب توابعی از t_x و t_y هستند .جعبه‌ها در کلاس‌های خود پیش‌بینی می‌شوند و برای محاسبه خطا، آنتروپی متقاطع دودویی (Binary Cross-Entropy) به‌عنوان تابع زیان استفاده می‌شود.سیستم در سه مقیاس متفاوت جعبه‌ها را پیش‌بینی می‌کند13 × 13, 26 × 26, 52 × 52 که برای اندازه ورودی 416×416 به‌کار می‌روند تا با feature map در مقیاس‌های مختلف ترکیب شده و اشیاء در اندازه‌های مختلف تشخیص داده شوند. برای تبدیل یک تصویر 416×416 به اندازه‌های مختلف، گام‌های حرکتی (strides) زیر استفاده می‌شوند:

8 پیکسل برای تبدیل تصویر به 52×52

16 پیکسل برای تبدیل تصویر به 26×26

32 پیکسل برای تبدیل تصویر به 13×13

52×52 برای شناسایی اشیای کوچک، 26×26 برای اشیای متوسط و 13×13 برای اشیای بزرگ استفاده می‌شود.

YOLOv4 architecture:

معماری YOLOv4

YOLOv4یک شبکه‌ی کاملاً کانولوشنی (CNN) است که شامل 110 لایه کانولوشنی می‌باشد. از میان این لایه‌ها، 66 لایه از نوع 1×1 و 44 لایه از نوع 3×3 هستند. شبکه از یک لایه‌ی ورودی کانولوشنی 3×3 با 32 فیلتر آغاز می‌شود که تصویری با اندازه 416×416 و 3 کانال (RGB) را دریافت می‌کند. لایه‌ی خروجی، که یک لایه‌ی کانولوشنی 1×1 با گام (stride) و پدینگ (padding)برابر 1 است، شامل 33 فیلتر می‌باشد.

اجزای اصلی معماری YOLOv4

Backbone

از مدل CSPDarknet53 برای استخراج ویژگی‌های عمیق تصاویر ورودی استفاده می‌شود.

Neck

SPP (Spatial Pyramid Pooling) به‌طور کارآمد میدان دید را افزایش می‌دهد.

PAN (Path Aggregation Network) ویژگی‌ها را در مقیاس‌های مختلف استخراج می‌کند.

Head

برای شناسایی اشیا از سر مدل YOLOv3 بهره گرفته می‌شود.

بهینه‌سازی مدل با استفاده از گرادیان نزولی مینی‌بچ همراه با مومنتوم انجام می‌شود و در لایه‌ی نهایی از تابع فعال‌سازی خطی استفاده شده است. برای ورودی با اندازه 416×416 و 3 کانال(RGB) ، مدل YOLOv4 دارای بیش از 60 میلیون پارامتر می‌باشد.

YOLOv4از تکنیک‌های مختلفی بهره می‌برد که به دو دسته‌ی اصلی تقسیم می‌شوند:

Bag of Freebies (BoF)

برای ستون فقرات(Backbone)

- افزایش داده‌ی موزاییکی(Mosaic Data Augmentation)

- نرمال‌سازیDropBlock
- فعال‌سازیMish
- اتصالات جزئی بین مرحله‌ای(Cross-Stage Partial Connections - CSP)

برای آشکارساز(Detector)

- نرمال‌سازیDropBlock
- افزایش داده‌ی موزاییکی(Mosaic Data Augmentation)
- آموزش خودمتخاصم(Self-Adversarial Training)
- استفاده از چندین لنگر (Anchor) برای یک حقیقت زمینی(Ground Truth)
- به‌کارگیری ابرپارامترهای بهینه(Optimal Hyperparameters)
- استفاده از اشکال تصادفی برای آموزش(Random Training Shapes)

Bag of Specials (BoS)

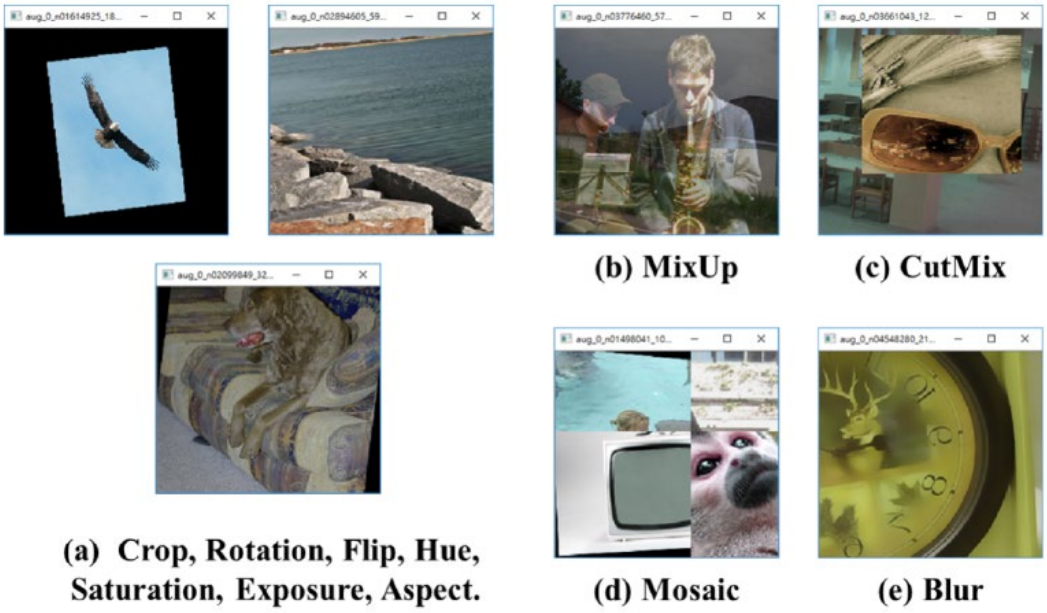
برای ستون فقرات:(Backbone)

- فعال‌سازیMish
- اتصالات جزئی بین مرحله‌ای(CSP)

برای آشکارساز(Detector)

- فعال‌سازیMish
- بلوک(SPP-block) SPP
- بلوک(PAN-block) PAN

این تکنیک‌ها در کنار یکدیگر باعث افزایش دقت و کارایی مدل در شناسایی اشیاء در تصاویر می‌شوند.



تکنیک‌های افزایش داده و بخش‌های مرتبط درYOLOv4

Mosaic

در این روش، چهار تصویر آموزشی با هم ترکیب می‌شوند. شکل بالا d

CutMix

در این تکنیک، دو تصویر ورودی به‌گونه‌ای ترکیب می‌شوند که اشیاء خارج از زمینه‌ی معمول آن‌ها قابل شناسایی باشند. شکلc

روش‌های افزایش داده دیگر:

روش‌هایی مانند MixUp و Blur نیز در شکل بالا نمایش داده شده‌اند.

(Self-Adversarial Training - SAT)

در این روش، ابتدا تصویر اصلی تغییر داده می‌شود؛ سپس شبکه‌ی عصبی برای شناسایی شیء موجود در تصویر اصلاح‌شده آموزش می‌یابد.

(Cross mini-Batch Normalization - CmBN)

در این روش، تنها آمار مربوط به مینی‌بچ‌های داخل یک بچ منفرد جمع‌آوری می‌شود.

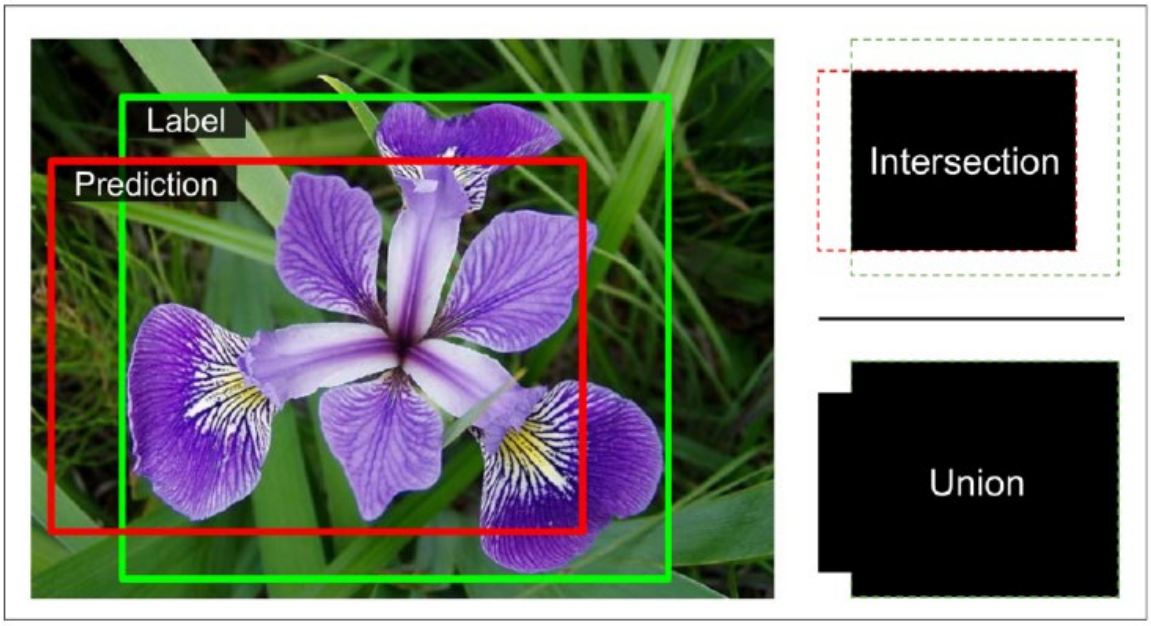
تابع هزینه برای YOLOv4

پیش از پرداختن به جزئیات تابع هزینه در YOLOv4 ، برخی از مفاهیمی که در این تابع به‌کار رفته‌اند، به‌صورت زیر توضیح داده میشود.

IoU (Intersection over Union)

این معیار، مساحت هم‌پوشانی بین باکس محدودکننده‌ی پیش‌بینی‌شده و باکس محدودکننده‌ی هدف (ground truth) را نسبت به مساحت اجتماع آن‌ها محاسبه می‌کند (رجوع به شکل 13). به عبارت دیگر، اگر B نشان‌دهنده باکس پیش‌بینی‌شده و باکس هدف باشد، $B \cap B_{gt}$ نواحی مشترک و $B \cup B_{gt}$ نواحی اجتماع را مشخص می‌کند. IoU یک معیار ارزیابی مقیاس‌ناپذیر است که به‌طور گسترده در شناسایی اشیا مورد استفاده قرار می‌گیرد.

$$\boldsymbol{IoU = \frac{B \cap B_{gt}}{B \cup B_{gt}}}$$



در این شکل،

$\boldsymbol{B \cap B_{gt}}$ نشان‌دهنده اشتراک نواحی بین باکس محدودکننده‌ی پیش‌بینی‌شده B و باکس محدودکننده‌ی حقیقت زمینی (ground truth bounding box) است.

$\boldsymbol{B \cup B_{gt}}$ ننشان‌دهنده اجتماع نواحی بین این دو باکس می‌باشد.

تابع هزینهDIoU-NMS

تابع هزینه DIoU (Distance IoU) نزدیکی بین باکس هدف و باکس پیش‌بینی‌شده را اندازه‌گیری می‌کند.

فرمول تابع هزینه DIoU به‌صورت زیر ارائه شده است:

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b, b_{gt})}{c^2}$$

در این فرمول:

b و b_{gt} به ترتیب مراکز هندسی (centroids) باکس پیش‌بینی‌شده B و باکس حقیقت زمینی B_{gt} هستند.

$d = \rho^2(b, b_{gt})$ فاصله بین مراکز باکس‌های B و B_{gt} را نشان می‌دهد c طول قطر کوچک‌ترین باکس محصورکننده‌ای است که هر دو باکس را در بر می‌گیرد

علاوه بر این، تکنیک غیربیشینه‌سازی (Non-Maximum Suppression - NMS) برای حذف باکس‌های غیرضروری که یک شیء را چندین بار شناسایی می‌کنند به کار می‌رود. در این روش، تنها باکس دارای بالاترین امتیاز اطمینان نگه‌داشته می‌شود. تابع NMS یک آرایه از باکس‌ها را دریافت کرده و بر اساس یک مقدار آستانه‌ای، میزان هم‌پوشانی آن‌ها را پردازش می‌کند.

Complete IoU (CIoU)

تابع هزینهCIoU ، نسخه توسعه‌یافته‌ای از DIoU Loss است که علاوه بر مساحت هم‌پوشانی و فاصله بین مراکز باکس‌ها، نسبت ابعاد آن‌ها را نیز در رگرسیون باکس محدودکننده در نظر می‌گیرد. این ویژگی باعث بهبود دقت و سرعت در شناسایی اشیاء می‌شود.

فرمول تابع هزینه CIoU به‌صورت زیر بیان شده است:

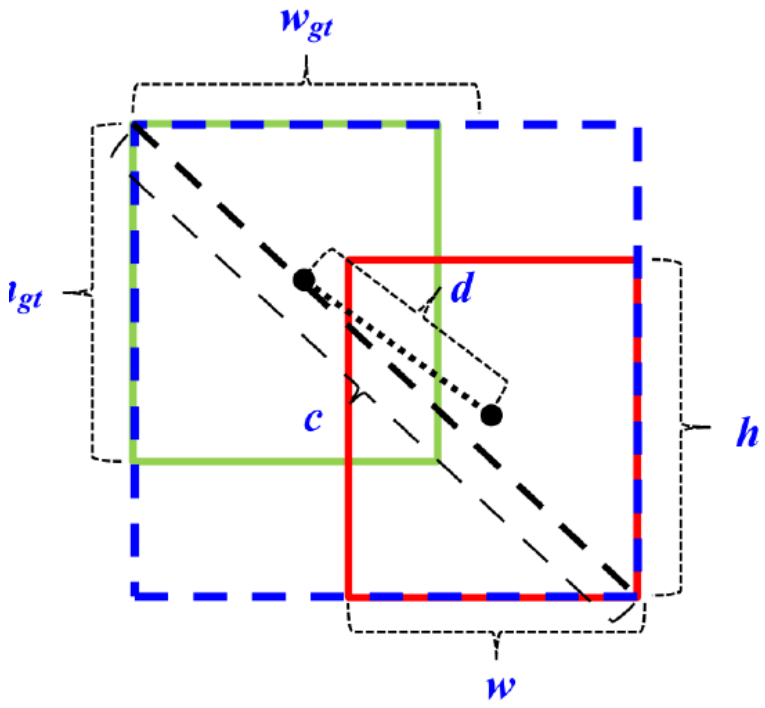
$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b_{gt})}{c^2} + \alpha u$$

$$u = (4/\pi^2) * \left(tan^{(-1)}(w_g t/h_g t) - tan^{(-1)}(w/h)\right)^2 and \alpha = u/((1 - IoU) + u')$$

توضیحات و نمادها

w_{gt} و h_{gt} به ترتیب عرض و ارتفاع باکس حقیقت زمینی (ground truth) هستند.

w و h نیز به ترتیب عرض و ارتفاع باکس پیش‌بینی‌شده می‌باشند.



همچنین:

u پارامتر نسبت ابعاد (aspect ratio parameter) است،

u' مشتق اول u می‌باشد،

α پارامتر موازنه‌ی مثبتی است که بر اساس DIoU loss تعیین شده است.

تابع هزینه کلی مدلYOLOv4

تابع هزینه‌ی کلی L برای YOLOv4 به‌صورت زیر تعریف می‌شود:

$$
\begin{aligned}
L =& 1 - IoU + \frac{\rho^2(\mathbf{b},\mathbf{b}_{gt})}{c^2} + \alpha u - \\
& \sum_{i=1}^{S^2}\sum_{j=1}^{B} I_{ij}^{obj}[\hat{C}_i log(C_i) + (1-\hat{C}_i) log(1-C_i) - \\
& \lambda_{noobj}\sum_{i=1}^{S^2}\sum_{j=1}^{B} I_{ij}^{obj}[\hat{C}_i log(C_i) + (1-\hat{C}_i) log(1-C_i) - \\
& \sum_{i=1}^{S^2}\sum_{j=1}^{B} I_{ij}^{obj} \sum_{c\in classes} [\hat{p}_i(cl) log(p_i(cl)) - (1-\hat{p}_i(cl)) log(1-p_i(cl))]
\end{aligned}
$$

تابع هزینه‌ی مدل YOLOv4 شامل سه بخش است:

هزینه‌ی رگرسیونی: خط اول معادله

هزینه‌ی اطمینان (Confidence Loss) : نمایان در خط‌های دوم و سوم معادله در این بخش، تابع هزینه از خطای آنتروپی متقاطع (cross-entropy error) استفاده می‌کند.

S^2 تعداد کل نقاط شبکه در تصویر ورودی را نشان می‌دهد.

B تعداد کل باکس‌های محدودکننده‌ی مرتبط با هر شبکه است.

λ_{noobj} پارامتری است که وزن هزینه‌ی اطمینان را تنظیم می‌کند.

I_{ij}^{obj} نشان می‌دهد که آیا شیئی در سلول i وجود دارد و آیا پیش‌بینی باکس محدودکننده‌ی j در سلول i مسئول این پیش‌بینی است. به عبارت دیگر، اگر شیئی در باکس محدودکننده‌ی j که توسط شبکه i تولید شده باشد، مقدار I_{ij}^{obj} برابر با 1 خواهد بود؛ در غیر این صورت، مقدار آن 0 در نظر گرفته می‌شود.

هزینه‌ی طبقه‌بندی(Classification Loss) : در این بخش نیز از خطای آنتروپی متقاطع استفاده می‌شود.

نمایانگر کلاسی است که به شناسایی هدف مربوط می‌شود.

$p_i(cl)$ احتمال واقعی تشخیص شیء از کلاس cl در شبکه i است.

$\hat{p}_i(cl)$ امتیاز احتمالی پیش‌بینی‌شده از کلاس cl می‌باشد.