



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)

## دانشکده مهندسی کامپیوتر

### گزارش کتبی کار مطالعاتی درس بینایی کامپیوتر

# مدل‌های تشخیص تک مرحله‌ای اشیاء

استاد درس

جناب آقای دکتر رضا صفابخش

نگارش

محسن عبادپور

نیمسال اول تحصیلی ۱۴۰۲-۱۴۰۱

۱۴۰۱/۱۱/۰۱

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## چکیده

تشخیص وجود اشیاء در یک صحنه و دسته‌بندی آن یک از مهم‌ترین اهداف و حوزه‌های فعال در مباحث بینایی کامپیوتر می‌باشد که کاربردهای عملیاتی و تحلیلی بسیاری بهمراه داشته و آن را به یکی از موضوعات داغ تحقیقاتی تبدیل نموده است بطوریکه مطالعه‌ی آن انگیزه‌ی توامی را در پی دارد. یکی از دو رویکرد اساسی تشخیص اشیاء که مبتنی بر سیستم بینایی انسان می‌باشد، مدل‌های تک مرحله‌ای می‌باشند که صرفا با یکبار پردازش تصویر ورودی اقدام به تشخیص و شناسایی اشیاء می‌کنند. مزیت مدل‌های مذکور در سرعت اجرایی بالا نهفته است اما در بحث دقت پیش‌بینی با چالش مواجه هستند. برای حل این مشکل، راه حل‌های متعددی نظری اصلاح تابع هزینه و صرف نظر از جعبه‌های مرزی پیش‌فرض ارائه شده و توانسته‌اند در رقابت با رویکرد دیگر به موفقیت‌های قابل توجهی دست یابند. در این گزارش مطالعاتی، اولین مدل تشخیص اشیاء مبتنی بر شبکه‌های عمیق معرفی و بررسی شده و در ادامه با بررسی چند پژوهش در راستای آن، بهبودهای ارائه شده برای افزایش دقت تشخیص و شناسایی اشیاء مورد مطالعه قرار گرفته است.

**کلمات کلیدی:** هرم ویژگی، شبکه‌های عمیق تشخیص اشیاء، مدل‌های تک مرحله‌ای، تابع هزینه

کانونی

## فهرست مطالب

۱	فصل یک مقدمه
۲	۱ - ۱- بینایی کامپیوتر و تشخیص اشیاء.....
۳	۱ - ۲- مدل‌های کلاسیک تشخیص اشیاء.....
۴	۱ - ۳- رویکردهای تشخیص اشیاء.....
۵	فصل دوم تشخیص اشیاء تک مرحله‌ای مبتنی بر شبکه‌های عمیق.....
۶	۶ - ۱- تاریخچه و پیدایش .....
۷	۶ - ۲- اولین ایده توسعه شبکه یولو.....
۸	۸ - ۳- تابع هزینه اعمالی.....
۹	۹ - ۴- گزارش نتایج .....
۱۰	فصل سوم بهبود تشخیص چندین شیء.....
۱۱	۱۰ - ۱- شبکه تشخیص اشیای اس اس دی .....
۱۲	۱۰ - ۲- چارچوب پیشنهادی .....
۱۳	۱۰ - ۳- محاسبه‌ی خطای خطا .....
۱۴	۱۰ - ۴- گزارش نتایج .....
۱۵	فصل چهارم اصلاح تابع هزینه .....
۱۶	۱۶ - ۱- شبکه‌ی تشخیص اشیای متراکم(رتینا) .....
۱۷	۱۶ - ۲- مشکل موجود .....
۱۸	۱۶ - ۳- راه حل پیشنهادی .....
۱۹	۱۶ - ۴- گزارش نتایج .....
۲۰	فصل پنجم تشخیص مرکز اشیاء.....
۲۱	۲۰ - ۱- مشکل موجود در جعبه‌های مرزی پیش‌فرض .....
۲۲	۲۰ - ۲- راهکار شبکه مرکز محور .....
۲۳	۲۰ - ۳- گزارش نتایج .....
۲۴	فصل ششم استفاده از پیکسل .....
۲۵	۲۵ - ۱- شبکه تشخیص شیء کاملا پیچشی .....
۲۶	۲۵ - ۲- تابع هزینه و آموزش .....
۲۷	۲۵ - ۳- گزارش نتایج .....
۲۸	فصل هفتم بهبود ارتباطات هرم ویژگی در تشخیص اشیاء .....
۲۹	۲۸ - ۱- چالش موجود در هرم ویژگی .....
۳۰	۲۸ - ۲- هرم ویژگی دو طرفه .....
۳۱	۲۸ - ۳- گزارش نتایج .....
۳۲	فصل هشتم جمع بندی و مراجع .....
۳۳	۳۲ - ۱- بحث .....



## فهرست اشکال

..... ۳	شکل ۱-۱: نمونه از تشخیص چندین شیء در یک تصویر
..... ۶	شکل ۱-۲: معماری شبکه عصبی پیچشی مورد استفاده در یولو
..... ۷	شکل ۲-۱: نمایش رابطه‌ی اشتراک بر اجتماع
..... ۸	شکل ۲-۲: جزئیات تنسور خروجی در شبکه یولو
..... ۱۱	شکل ۲-۳: مثال تشخیص اشیاء در تصاویر هنری با شبکه یولو
..... ۱۳	شکل ۳-۱: معماری شبکه اس‌اس‌دی و مقایسه‌ی آن با شبکه یولو
..... ۱۴	شکل ۳-۲: نمونه‌ای از سرکوب غیر حداکثری در تشخیص جعبه‌ی مرزی به اشیاء
..... ۱۷	شکل ۳-۳: نمونه‌هایی از تشخیص اشیاء توسط شبکه اس‌اس‌دی
..... ۲۲	شکل ۴-۱: معماری شبکه رتینا
..... ۲۳	شکل ۴-۲: نمودار بررسی تابع هزینه کانونی با تغییرات ضریب تاثیر
..... ۲۴	شکل ۴-۳: مقایسه سرعت و دقت شبکه رتینا
..... ۲۸	شکل ۵-۱: معماری شبکه تشخیص اشیای مرکز محور
..... ۳۰	شکل ۵-۲: بهبود تشخیص اشیای بسیار کوچک در شبکه‌ی مرکز محور
..... ۳۲	شکل ۶-۱: ساختار شبکه تشخیص شیء کاملاً پیچشی
..... ۳۴	شکل ۶-۲: چگونگی اختصاص یک جعبه‌ی مرزی به یک پیکسل در شبکه تشخیص شیء کاملاً پیچشی
..... ۳۴	شکل ۶-۳: نتیجه‌ی اعمال معیار مرکزیت به ازای پیکسل‌ها در شبکه تشخیص شیء کاملاً پیچشی
..... ۳۸	شکل ۷-۱: طرحواره هرم ویژگی
..... ۳۹	شکل ۷-۲: طرحواره هرم ویژگی دو طرفه
..... ۴۰	شکل ۷-۳: معماری شبکه تشخیص کارآمد اشیاء

## فهرست جداول

جدول ۲-۱: مقایسه شبکه یولو با شبکه‌های تشخیص اشیاء بلادرنگ ..... ۱۰
جدول ۲-۲: مقایسه شبکه یولو با شبکه‌های غیر بلادرنگ ..... ۱۰
جدول ۲-۳: مقایسه شبکه یولو روی تصاویر هنری ..... ۱۱
جدول ۳-۱: مقایسه دقت شبکه اس‌اس‌دی با شبکه آرسی‌ان‌ان سریع، آرسی‌ان‌ان سریع‌تر و یولو ..... ۱۶
جدول ۳-۲: مقایسه سرعت شبکه اس‌اس‌دی با شبکه آرسی‌ان‌ان سریع، آرسی‌ان‌ان سریع‌تر و یولو ..... ۱۷
جدول ۵-۱: جدول مقایسه‌ی شبکه‌ی تشخیص اشیای مرکز محور با سایر شبکه‌ها ..... ۳۰
جدول ۶-۱: مقایسه عملکرد شبکه‌ی تشخیص شیء کاملاً پیچشی با سایر شبکه‌های تک و دو مرحله‌ای ..... ۳۶
جدول ۷-۱: مقایسه‌ی عملکرد و جزئیات اجرایی شبکه تشخیص کارآمد اشیاء با سایر شبکه‌ها ..... ۴۱

## فهرست روابط

۹	رابطه (۱-۲) : تابع هزینه شبکه یولو.....
۱۵	رابطه (۱-۳) : تابع هزینه اعمالی در شبکه اس اس دی .....
۲۰	رابطه (۱-۴) : تابع هزینه آنتروپی متقابل .....
۲۱	رابطه (۴-۲) : بازنویسی تابع هزینه آنتروپی متقابل.....
۲۱	رابطه (۴-۳) : اصلاح تابع هزینه آنتروپی متقابل و وزن دار کردن آن نسبت به کلاس .....
۲۲	رابطه (۴-۴) : تابع هزینه کانونی.....
۲۹	رابطه (۱-۵) : تابع هزینه شبکه تشخیص اشیای مرکز محور .....
۳۶	رابطه (۶-۱): تابع هزینه شبکه تشخیص شیء کاملا پیچشی .....

## فهرست علائم و اختصارات

عنوان یا علامت اختصاری	عنوان کامل
IoU	رابطه‌ی اشتراک بر اجتماع دو ناحیه
C	برچسب کلاس‌های مجموعه داده
y	برچسب هر نمونه آموزشی
$t^*$	جعبه‌ی مرزی هدف
p	احتمال تخصیص برچسب کلاس به نمونه
CE(p,y)	تابع هزینه آنتروپی متقابل
L <sub>loc</sub> /L <sub>reg</sub>	خطای جعبه‌ی مرزی پیش‌بینی شده
L <sub>cls</sub> /L <sub>det</sub> /L <sub>conf</sub>	خطای دسته‌بندی اشیاء
a <sub>t</sub>	ضریب تابع هزینه آنتروپی متقابل به ازای نمونه‌های هر کلاس
FL	تابع هزینه کانونی(مرکزی)
N <sub>pos</sub>	تعداد پیکسل‌های مثبت در جعبه‌ی مرزی پیشنهادی
W <sub>i</sub> /H <sub>i</sub>	طول و عرض جعبه‌ی مرزی پیشنهادی i ام
L <sub>off</sub>	خطای حاصل از اختلاف جعبه‌ی مرزی offset (offset مخفف off)
N	تعداد نمونه‌ی آموزشی

## فصل يك مقدمه

## ۱ - ۱- بینایی کامپیوتر و تشخیص اشیاء

بینایی کامپیوتر<sup>۱</sup> به عنوان یکی از قدیمی‌ترین و در عین حال اساسی‌ترین زمینه‌های تحقیقاتی در عرصه هوش مصنوعی شناخته می‌شود که فعالیت‌های متعددی را در بر می‌گیرد. وظایف<sup>۲</sup> بینایی کامپیوتر شامل فعالیت‌های سطح پایین<sup>۳</sup> و سطح بالا<sup>۴</sup> می‌باشد که مطالعات در آن جریان دارد؛ فعالیت‌های سطح پایین اموراتی نظیر تشخیص نقاط کلیدی<sup>۵</sup>، اصلاح نور و روشنایی<sup>۶</sup>، لبه‌یابی<sup>۷</sup> و... را شامل شده و مقدمات فعالیت‌های سطح بالا را فراهم می‌سازد و به تنها‌ی آورده‌ای مفید برای اهداف نهایی بینایی محسوب نمی‌شود.

فعالیت‌های سطح بالا اهداف و مقاصدی قابل درک می‌باشند و نتیجه‌ی این فعالیت‌ها مستقیماً دنبال می‌شود که از آن می‌توان به دسته‌بندی<sup>۸</sup>، تقطیع<sup>۹</sup>، تشخیص اشیاء<sup>۱۰</sup> و... اشاره نمود. در حالت کلی و در تصاویر عمومی بندرت پیش می‌آید که صرفاً یک شیء در تصویر وجود داشته و هدف دسته‌بندی آن باشد؛ لذا در اکثر اوقات چندین شیء در تصاویر وجود دارد. از این رو، بایستی ابتدا ناحیه‌ی اشیای موجود استخراج و مکان‌یابی<sup>۱۱</sup> اتفاق افتد و سپس اشیای مورد نظر دسته‌بندی شوند؛ از فرآیند مذکور تحت عنوان تشخیص اشیاء در تصاویر یاد می‌شود و از دیرباز پژوهش در این حوزه وجود داشته و حتی هم اکنون نیز با پیدایش شبکه‌های عصبی عمیق<sup>۱۲</sup> مسئله‌ای چالشی محسوب می‌شود.

تشخیص درست اشیاء در تصاویر بسیار حائز اهمیت است زیرا اطلاعات مهمی را فراهم می‌کند که می‌توان به وجود یا عدم وجود شیء، تعداد اشیاء، موقعیت و ارتباط بین اشیاء، برچسب<sup>۱۳</sup> اشیاء و... اشاره نمود. هر یک از اطلاعات یاد شده به تنها‌ی یا با یکدیگر می‌توانند مورد استفاده قرار گیرند که دریایی از مسائل را شامل می‌شود

<sup>1</sup> Computer Vision

<sup>2</sup> Tasks

<sup>3</sup> Low-Level

<sup>4</sup> High-Level

<sup>5</sup> Key Points

<sup>6</sup> Light Correction

<sup>7</sup> Edge Detection

<sup>8</sup> Classification

<sup>9</sup> Segmentation

<sup>10</sup> Object Detection

<sup>11</sup> Localization

<sup>12</sup> Deep Neural Network

<sup>13</sup> Label

که از آن می‌توان به اموراتی نظری تشخیص گروهی فعالیت، تفسیر و تحلیل تصویر اشاره نمود. نمونه‌ای از تشخیص چندین اشیاء در شکل قابل مشاهده است.



شکل ۱-۱: نمونه از تشخیص چندین شیء در یک تصویر

### ۱ - ۲ - مدل‌های کلاسیک تشخیص اشیاء

قبل از ظهرور شبکه‌های عصبی عمیق مدل‌های کلاسیک بینایی کامپیوتر برای تشخیص اشیاء مورد استفاده قرار می‌گرفتند که امروزه استفاده از آن کم‌رنگ شده است. در مدل‌های کلاسیک، ابتدا تصاویری از اشیای مورد نظر پردازش قرار گرفته و از آن نقاط کلیدی و ویژگی<sup>۱۴</sup> توسط الگوریتم هایی نظری سیفت<sup>۱۵</sup> [۱] استخراج می‌شود. سپس به ازای تصویر ورودی و تک تک اشیای موجود اولیه، فرآیند تطبیق<sup>۱۶</sup> نقاط کلیدی و ویژگی‌ها صورت گرفته و شیء مفروض جستجو می‌گردد. این مراحل با وجود اینکه نتایجی مناسبی را می‌تواند حاصل کند، اما نقاط ضعف متعددی را به همراه دارد که می‌توان به کند بودن، پرهزینه بودن و عدم مقاومت در برابر تغییرات ظاهری شیء اشاره نمود.

تغییرات ظاهری شیء چالشی است که بیشترین نقص را به مدل‌های کلاسیک تشخیص اشیاء وارد می‌کند چرا که ماهیت اشیاء مبنای تشخیص قرار نگرفته و صرفاً جستجوی رخ می‌دهد؛ برای مثال در شکل ۱-۱ اگر نوع و رنگ گوشی تلفن همراه تغییر یابد، مدل‌های کلاسیک دیگر به خوبی نخواهند توانست تشخیص اشیاء را انجام دهند.

### ۱ - ۳ - رویکردهای تشخیص اشیاء

در بخش ۱-۱ فرآیند تشخیص اشیاء مورد اشاره قرار گرفته شد که در این بخش دقیق‌تر بررسی خواهد شد. تشخیص اشیاء از دو مرحله‌ی زیر تشکیل شده است:

<sup>14</sup> Feature

<sup>15</sup> SIFT (Scale-Invariant Feature Transform)

<sup>16</sup> Matching

(۱) مکان یابی شیء: در این مرحله، هدف جست و جو و یافتن موقعیت، ابعاد و ناحیه‌ی هر گونه شیء موجود در تصویر است که با عنوان نواحی مطلوب<sup>۱۷</sup> از آن یاد می‌شود. این مرحله دارای چالش‌های سختی می‌باشد که می‌توان به تغییر ویژگی‌های مکانی و شرایط کنترل نشده‌ی محیط اشاره نمود.

(۲) دسته‌بندی و شناسایی شیء: پس از استخراج نواحی مطلوب از تصویر ورودی، هر یک از آن بصورت جداگانه به دسته‌بند مورد نظر داده می‌شود تا فرآیند شناسایی انجام پذیرفته و برچسب شیء موجود در آن ناحیه حاصل شود.

با توجه به دو مرحله‌ی فوق در تشخیص اشیا، می‌توان دو رویکرد برای آن ترسیم نمود. رویکرد اول که با عنوان تشخیص دو مرحله‌ای اشیاء<sup>۱۸</sup> شناخته می‌شود بدین گونه عمل می‌کند که یکبار تصویر ورودی پردازش و محل اشیاء مکانی‌بایی شده و نواحی مطلوب از تصویر استخراج می‌شود. در گام مجازی بعدی، هر یک از نواحی مطلوب دوباره پردازش شده و عمل دسته‌بندی روی آن انجام می‌پذیرد و در نهایت برچسب آن مشخص می‌شود. این رویکرد تقریباً اساس مدل‌های کلاسیک بوده و برخی از مدل‌های مبتنی بر شبکه‌های عصبی عمیق نیز از این رویکرد بهره جسته‌اند که می‌توان به شبکه‌های آرسی‌ان‌ان<sup>[۱۹]</sup> و اف‌پی‌ان<sup>[۲۰]</sup> [۳] اشاره نمود.

در رویکرد دوم، برخلاف رویکرد اول دو مرحله‌ی استخراج نواحی مطلوب و شناسایی بصورت همزمان اتفاق افتاده و صرفاً یکبار فرآیند پردازش انجام می‌پذیرد. این رویکرد که عمر کوتاه‌تری نسبت به رویکرد اول دارد، توانسته است به نتایج مطلوب در حالت کلی دست‌یابد. برای این رویکرد نیز شبکه‌های عصبی عمیق متعددی نظیر یولو<sup>[۲۱]</sup> [۴] و اس‌اس‌دی<sup>[۲۲]</sup> [۵] معرفی و پیشنهاد شده است؛ در ادامه‌ی این کار مطالعاتی، شبکه‌های عصبی عمیق چند سال اخیر و مبتنی بر رویکرد دوم مورد بررسی قرار خواهد گرفت.

<sup>17</sup> Regions of interests

<sup>18</sup> Two Stage Object Detection

<sup>19</sup> R-CNN (Regions with CNN features)

<sup>20</sup> Feature Pyramid Networks

<sup>21</sup> YOLO (You only look once)

<sup>22</sup> SSD (Single Shot Multibox Detector)

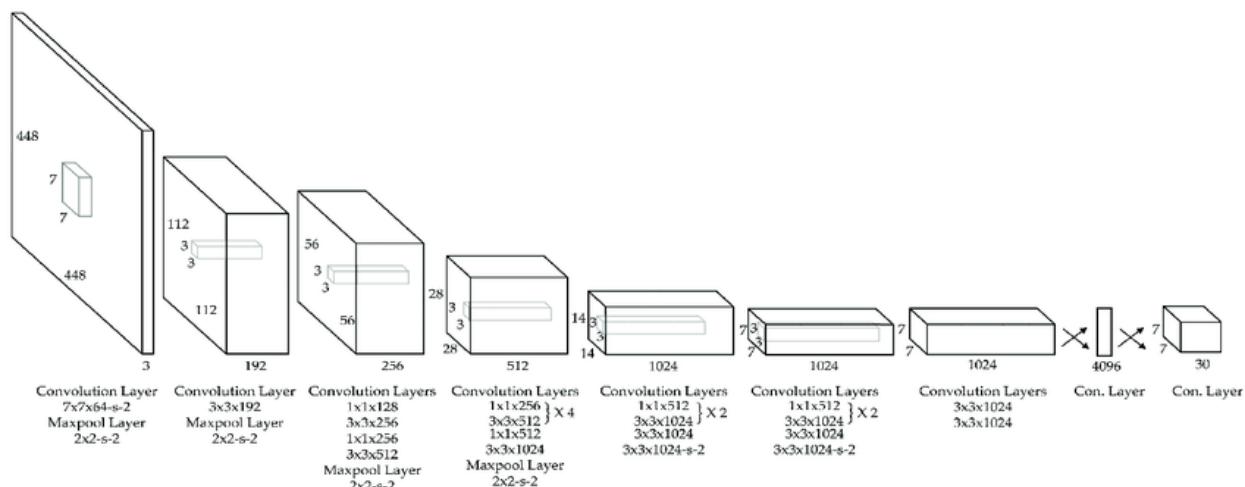
## فصل دوم تشخیص اشیاء تک مرحله‌ای مبتنی بر شبکه‌های عمیق

## ۲ - ۱ - تاریخچه و پیدایش

شبکه YOLO<sup>۲۳</sup> [۴] در ابتدای سال ۲۰۱۶ و تقریباً به عنوان اولین شبکه‌ی تشخیص شیء مبتنی بر شبکه‌های عصبی ارائه شده و در صدد تشخیص به صورت تک مرحله‌ای می‌باشد و عمدۀ تمرکز آن قابلیت بلادرنگ<sup>۲۴</sup> توأم با دقت بالا است بطوریکه بتوان فرآیند مذکور را در سطح کاربرد عملیاتی نمود. این شبکه با بکارگیری شبکه‌های مرسوم عصبی پیچشی<sup>۲۵</sup> [۶] در قسمت های مختلف تصویر و ترکیب نتایج هر یک با دیگری اقدام به تشخیص اشیا می‌کند. معماری و ساختار این شبکه در بخش ۲-۲ مورد اشاره قرار گرفته و در بخش ۲-۳ تابع هزینه و خطای اعمالی در گام آموزشی بررسی خواهد شد و در نهایت در بخش ۲-۴ نیز نتایج و مقایسه آن با سایر شبکه‌های تشخیص اشیاء انجام خواهد شد.

## ۲ - ۲ - اولین ایده توسط شبکه YOLO

در شبکه YOLO<sup>۴</sup> [۴] به عنوان اولین معماری پیشنهادی برای تشخیص اشیای تک مرحله‌ای مبتنی بر شبکه‌های عصبی علاوه بر سرعت و دقت بالا در تشخیص اشیای موجود در تصویر، توجه به جزئیات نیز انجام پذیرفته است. در این شبکه ابتدا تصویر ورودی به وضوح ۴۴۸\*۴۴۸<sup>۲۶</sup> تغییر سایز داده می‌شود تا بتوان از تمامی اطلاعات و جزئیات موجود در تصویر رنگی استفاده نمود. حال تصویر مفروض به یک شبکه‌ی عصبی پیچشی ورودی داده می‌شود که معماری آن در شکل ۱-۲ قابل مشاهده است.



شکل ۱-۲: معماری شبکه عصبی پیچشی مورد استفاده در YOLO [۴]

<sup>23</sup> YOLO (You Only Look Once)

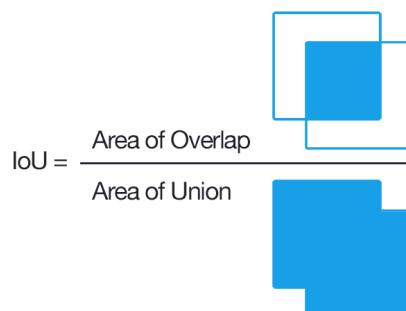
<sup>24</sup> Real-Time

<sup>25</sup> Convolutional Neural Network

<sup>26</sup> Resolution

شبکه عصبی پیچشی مورد استفاده از ۲۴ لایه‌ی پیچشی برای استخراج ویژگی استفاده شده و در انتهای شبکه نیز دو لایه‌ی تماماً متصل<sup>۲۷</sup> قرار داده شده است. ورودی شبکه یک تصویر سه کاناله‌ی رنگی باوضوح [۷] PASCAL VOC مجموعه داده‌ی ۴۴۸\*۴۴۸ بوده و خروجی آن نیز یک تنسور<sup>۲۸</sup> با ابعاد ۷\*۷\*۳۰ برای مجموعه داده‌ی ۷\*۷\*۳۰ می‌باشد. در نسخه یولوی سریع<sup>۲۹</sup> برای سرعت بخشیدن به شبکه و تشخیص اشیاء، به جای ۲۴ لایه‌ی پیچشی از ۹ لایه‌ی پیچشی استفاده شده است. بررسی وجود شیء در شبکه‌ی یولو به ازای تصویر ورودی بصورت نواحی محلی مد نظر گرفته می‌شود بگونه‌ای که تصویر در قالب بلوک‌های ۶۴\*۶۴ فرض شده و وجود شیء در بلوک‌ها هدف قرار داده می‌شود. از جایی که هم طول و هم عرض تصویر ۴۴۸ پیکسل می‌باشد، تعداد بلوک‌های حاصل برابر با  $7 \times 7 = 49$  (۶۴\*۶۴) خواهد بود؛ لذا در خروجی شبکه نیز ۷\*۷ نشان‌دهنده‌ی هر یک از این بلوک‌ها می‌باشد. به ازای هر بلوک وجود دو شیء و پیش‌بینی دو جعبه‌ی مرزی<sup>۳۰</sup> برای آن بررسی می‌شود. برای نمایش هر جعبه‌ی مرزی به پنج مقدار نیاز است که این پنج مقدار به ترتیب نشان دهنده‌ی ضریب اطمینان وجود شی<sup>۳۱</sup> در بلوک، مختصات مرکز، طول و عرض جعبه‌ی مرزی می‌باشد.

ضریب اطمینان از طریق رابطه اشتراک بر اجتماع<sup>۳۲</sup> محاسبه می‌شود بطوریکه اشتراک جعبه‌ی مرزی پیش‌بینی شده و جعبه‌ی مرزی مفروض یا یکدیگر حساب شده و تقسیم بر اجتماع شان می‌شود؛ مقدار حاصل یک عدد اعشاری بین صفر الی یک خواهد بود و هر چه به عدد یک نزدیکتر باشد، همپوشانی بیشتر رخ داده است که مطلوب مد نظر می‌باشد. طرحواره اشتراک بر اجتماع در شکل ۱-۲ قابل مشاهده است.



شکل ۱-۲: نمایش رابطه‌ی اشتراک بر اجتماع

<sup>27</sup> Fully-Connected

<sup>28</sup> Tensor

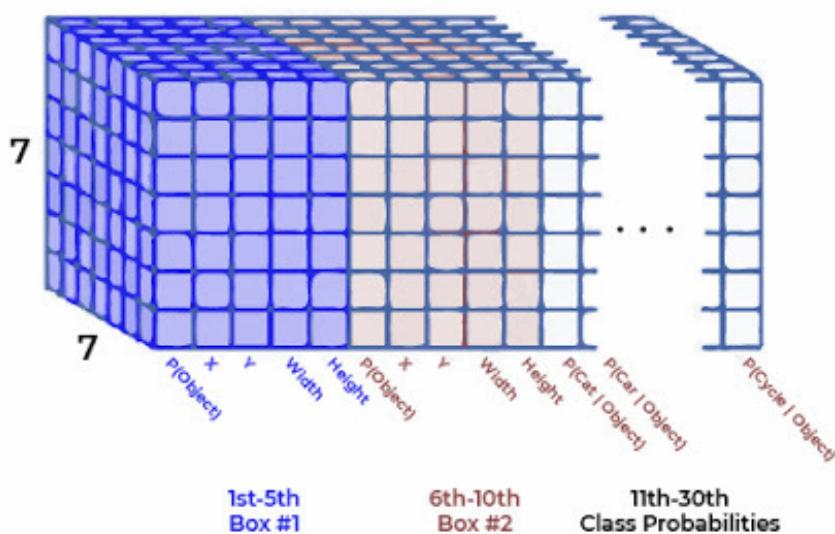
<sup>29</sup> Fast YOLO

<sup>30</sup> Bounding Box

<sup>31</sup> Confidence

<sup>32</sup> Intersection over Union(IoU)

برای هر کدام از بلوک‌های مفروض، بایستی احتمال وجود هر یک از کلاس‌های مجموعه داده نیز حاصل شود که در مجموعه داده‌ی PASCAL VOC [۷] برابر با ۲۰ می‌باشد. لذا برای هر بلوک بایستی ۳۰ مقدار حاصل شود که به ترتیب ۵ مقدار اول اطلاعات مربوط به تشخیص جعبه‌ی مرزی اول، ۵ مقدار دوم اطلاعات مربوط به جعبه‌ی مرزی دوم و ۲۰ مقدار بعدی هر یک احتمال تخصیص کلاس‌ها را نشان می‌دهد. بصورت کلی تر می‌توان نشان داد اگر تعداد بلوک‌های مفروض تصویر ورودی برابر با  $S^*S$  بوده و تعداد جعبه‌ی مرزی مد نظر برای پیش‌بینی در هر بلوک برابر با  $B$  باشد و تعداد کلاس قابل پیش‌بینی نیز معادل با  $C$  باشد، ابعاد خروجی شبکه‌ی یولو برابر با  $(S^*S^*(5B+C))$  خواهد بود؛ طرحواره‌ی مربوط به ابعاد تنسور خروجی در شکل ۲-۲ قابل مشاهده است.



شکل ۲-۳: جزئیات تنسور خروجی در شبکه یولو

### ۲ - ۳ - تابع هزینه اعمالی

حال اگر عمیق‌تر به روند تخمین جعبه‌های مرزی توجه کنیم، خواهیم دید که شبکه یولو در صدد تخمین مختصات مرکز و ابعاد آن می‌باشد که یک مسئله‌ی رگرسیون<sup>۳۳</sup> می‌باشد چرا که تلاش می‌کند مقادیر را بر اساس

<sup>۳۳</sup> Regression

میانگین مربعات خطای<sup>۳۴</sup> پیش‌بینی کند. برای نشان دادن بهتر این نکته، تابع هزینه‌ی این شبکه را بر اساس رابطه ۱-۲ بررسی می‌کنیم؛ عبارت اول تابع هزینه، خطای مختصات مرکز جعبه‌ی مرزی تخمین زده را بصورت رگرسیونی و بر اساس میانگین مربعات خطای محاسبه می‌کند. عبارت دوم نیز خطای طول و عرض جعبه‌ی مرزی تخمین زده را بصورت رگرسیونی و بر اساس میانگین مربعات خطای محاسبه می‌کند. عبارت سوم و چهارم نیز ضریب اطمینان وجود شیء را در دو حالت وجود و عدم وجود بدست می‌آورد و در نهایت خطای احتمال کلاس پیش‌بینی شده اضافه می‌شود.

$$\begin{aligned}
 & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
 \end{aligned}$$

[۱-۲] : تابع هزینه شبکه یولو

وجود عدد یک در عبارت‌های خطای نشان دهنده‌ی وجود شیء در جعبه‌ی مرزی و بلوک پردازشی به ازای تصویر ورودی می‌باشد و اگر شیء وجود نداشته باشد، با صفر جایگذین می‌شود.

## ۲ - ۴ - گزارش نتایج

خروجی و عملکرد شبکه‌های تشخیص اشیاء را می‌توان بر اساس دو معیار سرعت و دقیقت تحلیل و مقایسه نمود. تاکید شبکه یولو<sup>۴</sup> بصورت توانمند بر هر دو معیار مذکور می‌باشد اما توانسته است در سرعت عملکرد بهتری

<sup>۳۴</sup> MSE (Mean Squared Error)

## فصل دوم: تشخیص اشیاء تک مرحله‌ای مبتنی بر شبکه‌های عمیق

را از خود نشان دهد. معیار سرعت بر اساس تعداد فریم پردازشی در یک ثانیه<sup>۳۵</sup> و معیار دقت نیز بر اساس ام‌اپی<sup>۳۶</sup> گزارش شده است. جدول ۲-۱ مقایسه شبکه یولو را با شبکه‌های تشخیص اشیاء بلاذرنگ نشان می‌دهد.

جدول ۲-۱: مقایسه شبکه یولو با شبکه‌های تشخیص اشیاء بلاذرنگ [۴]

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	<b>155</b>
YOLO	2007+2012	<b>63.4</b>	45

همانطور که قابل مشاهده است، شبکه یولو و نسخه سریع آن بین شبکه‌های تشخیص اشیای بلاذرنگ توانسته است به دقت بیش از دو برابری و به سرعت بیش از ۱.۵ برابر دست یابد که در حوزه‌ی تشخیص اشیا و زمان خود یک انقلاب محسوب می‌شود. در جدول ۲-۲ نیز مقایسه شبکه یولو با شبکه‌های غیر بلاذرنگ آورده شده است. قابل مشاهده است که با وجود دقت اندکی کمتر نسبت به بهترین شبکه‌های مبتنی بر آرسی‌ان‌ان [۲]، یولو سرعتی تا ۴۰ برابر بهتر را ارائه می‌کند.

جدول ۲-۲: مقایسه شبکه یولو با شبکه‌های غیر بلاذرنگ [۴]

Less Than Real-Time	Train	mAP	FPS
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

در مقایسه‌ی چالشی دیگری اگر تشخیص اشیاء در تصاویر هنری و تصنیعی را مد نظر قرار دهیم، خواهیم دید که دقت شبکه یولو بسیار بیشتر از شبکه‌ی رقیب یعنی آرسی‌ان‌ان [۲] می‌باشد. برای مقایسه مذکور از دو

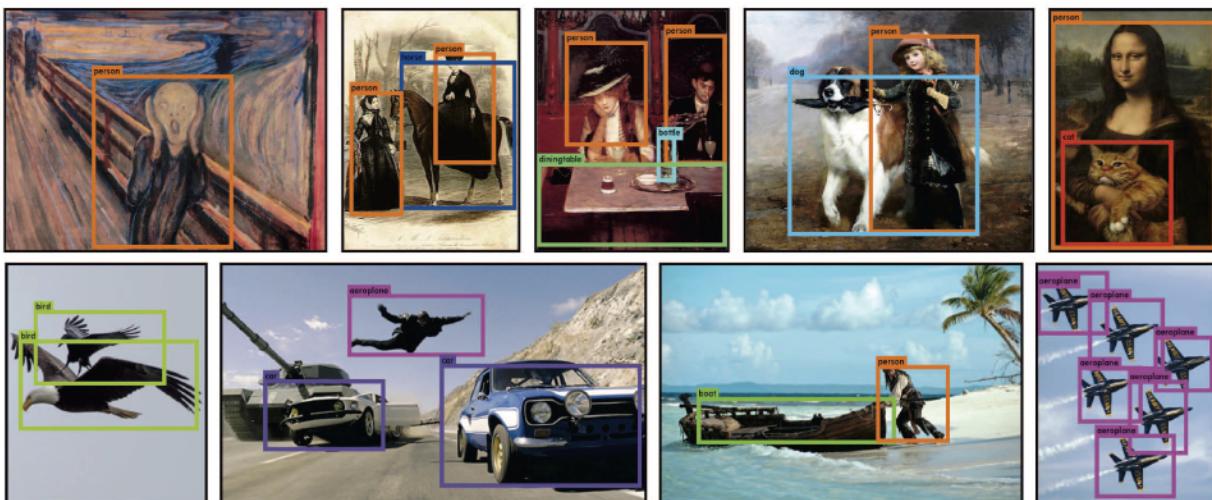
<sup>35</sup> FPS (Frame Per Second)

<sup>36</sup> mAP (mean Average Precision)

مجموعه داده Picasso [۸] و People-Art استفاده شده و نتایج حاصل در جدول ۳-۲ قابل مشاهده است. برای مثال بصری از تشخیص اشیاء در تصاویر هنری توسط شبکه یولو شکل ۴-۲ آورده شده است.

جدول ۳-۲: مقایسه شبکه یولو روی تصاویر هنری [۴]

	VOC 2007 AP	Picasso AP Best $F_1$		People-Art AP
	AP	Best $F_1$	AP	AP
YOLO	59.2	53.3	0.590	45
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	



شکل ۱-۲: مثال تشخیص اشیاء در تصاویر هنری با شبکه یولو [۴]

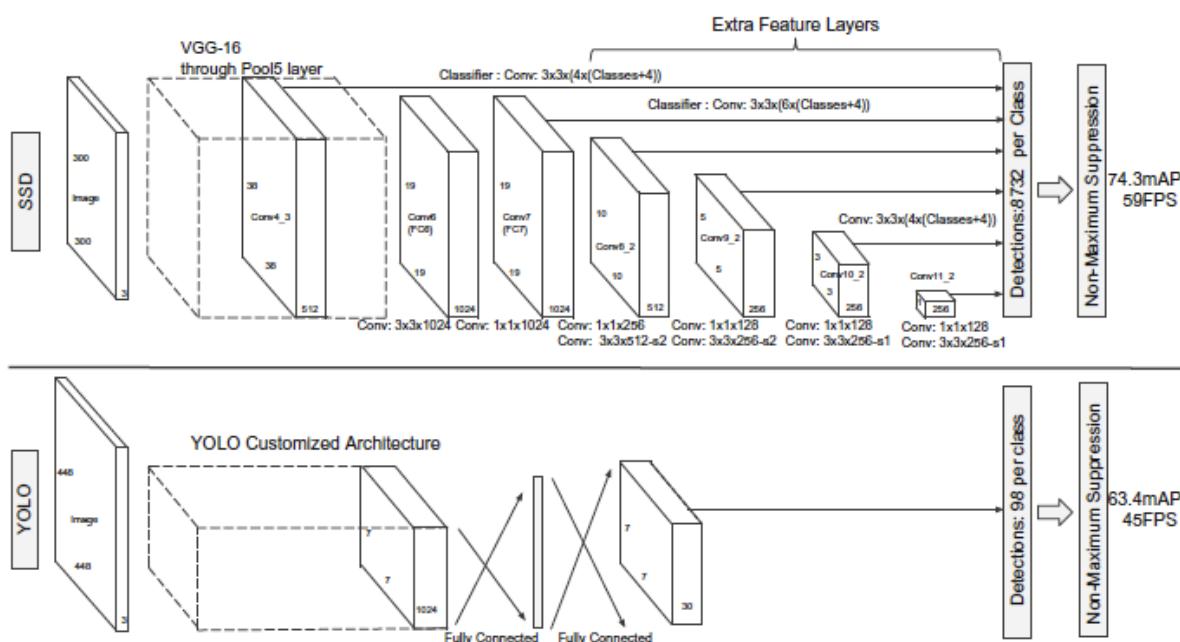
## **فصل سوم بهبود تشخیص چندین شیء**

### ۳ - ۱ - شبکه تشخیص اشیای اس اس دی

شبکه اس اس دی<sup>۳۷</sup> [۵] در ادامه سال ۲۰۱۶ و پس از شبکه یولو [۴] تقریباً به عنوان دومین شبکه تشخیص شیء مبتنی بر شبکه‌های عصبی ارائه شده و در صدد تشخیص به صورت تک مرحله‌ای می‌باشد و عمله تمرکز آن قابلیت تشخیص چند جعبه مرزی است بطوریکه بتوان فرآیند چالشی مذکور را به نحو مناسبی پاسخ داد. این شبکه فرآیندهای تشخیص اشیاء را در یک شبکه عصبی گنجانده و لذا پیچیدگی و حجم پردازش‌های مورد نیاز کاهش داشته است. چارچوب پیشنهادی این شبکه در بخش ۲-۳ مورد اشاره قرار گرفته و در بخش ۳-۳ تابع هزینه‌ی آن بررسی خواهد شد و در نهایت در بخش ۴-۳ نیز مقایسه آن با سایر شبکه‌های تشخیص اشیاء انجام خواهد پذیرفت.

### ۳ - ۲ - چارچوب پیشنهادی

ساختار شبکه اس اس دی مبتنی بر یک شبکه عصبی پیچشی جلو-رو<sup>۳۸</sup> [۶] می‌باشد. معماری شبکه اس اس دی و مقایسه‌ی آن با شبکه یولو در شکل ۱-۳ قابل مشاهده است.



شکل ۳: معماری شبکه اس اس دی و مقایسه‌ی آن با شبکه یولو [۵]

<sup>37</sup> SSD (Single Shot multibox Detector)

<sup>38</sup> Feed-Forward

معماری شبکه اس اس دی با تکیه بر جست و جوی اشیای موجود در مقیاس های <sup>۳۹</sup> مختلف تشکیل شده است و در ابتدای خود از شبکه پیچشی VGG-16 [۶] برای استخراج ویژگی استفاده می کند که در آن ابعاد ورودی شبکه برابر با  $300 * 300$  در نظر گرفته شده است. روی خروجی VGG-16 [۶] از شش لایه پیچشی متوالی برای استخراج صفحات ویژگی <sup>۴۰</sup> استفاده شده و برای شناسایی به کار گرفته می شود و مقادیر هر یک از این شش لایه به انتهای شبکه انتقال داده می شود تا بدین صورت بتوان پیش بینی های انجام شده در مقیاس های مختلف را در دست داشته و برای تعیین جعبه های مرزی استفاده نمود که در نهایت امکان پیش بینی و بررسی ۸۷۳۲ جعبه های مرزی را در مقیاس های گوناگون فراهم می آورد. لذا می توان این گونه خلاصه نمود که در این معماری از لایه های پیچشی برای تخمین و پیش بینی جعبه های مرزی استفاده می شود و خروجی کلیشه <sup>۴۱</sup> سنجش می شود و این در حالی است که در شبکه یولو لایه های تماماً متصل این وظیفه را بر عهده دارد.

تعداد ۸۷۳۲ جعبه های مرزی حاصل شده تعداد بسیار زیادی بوده و همه ای آنان جعبه های مرزی اشیای موجود در تصویر نمی باشد؛ فارغ از اینکه تعداد خیلی زیادی از آن ها با یکدیگر همپوشانی بالایی دارند. برای رفع این چالش از سرکوب مقادیر غیر حداکثری <sup>۴۲</sup> استفاده می شود که اگر این مورد انجام نپذیرد یک شی توسط چندین جعبه های مرزی می تواند احاطه شود که امری بدینها اشتباه است. نمونه ای این مورد در شکل ۲-۳ قابل مشاهده است که قسمت های مختلفی از خودروی موجود توسط جعبه های مرزی متفاوتی مشخص شده و هر یک را به تنها یک خودرو تشخیص داده است. حال اگر با اعمال یک حد آستانه جعبه های مرزی با احتمال کمتر را نادیده گرفته و سرکوب نماییم، این چالش رفع خواهد شد.



شکل ۳-۲: نمونه ای از سرکوب غیر حداکثری در تشخیص جعبه های مرزی به اشیاء

<sup>39</sup> Sacle

<sup>40</sup> Feature maps

<sup>41</sup> Kernel

<sup>42</sup> Non maximum suppression

### ۳ - ۳ - محاسبه‌ی خطای خطا

برای تابع هزینه در شبکه اس‌اس‌دی [۵] از رابطه ۱-۳ استفاده شده است. این تابع هزینه مشتمل بر محاسبه و مجموع دو خطای می‌باشد.

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

رابطه (۱-۳) : تابع هزینه اعمالی در شبکه اس‌اس‌دی [۵]

در تابع هزینه اعمالی، دو عبارت اصلی دیده می‌شود؛ عبارت اول ( $L_{loc}$ ) خطای مربوط به پیش‌بینی جعبه‌ی مرزی را نشان می‌دهد که وزنی از آن تاثیر داده می‌شود. دلیل استفاده از وزن این نکته است که بصورت عددی مقدار خطای تخمین جعبه بالاتر از خطای تخمینی برچسب کلاسی می‌باشد و می‌تواند آن را تحت تاثیر منفی خود قرار دهد و شبکه در امر یادگیری تشخیص برچسب موفق نباشد لذا از وزن دار آن استفاده می‌شود. پارامترهای استفاده شده‌ی ۱ و  $g$  نیز به ترتیب جعبه‌ی مرزی تخمینی و حقیقت مبنا<sup>۴۳</sup> می‌باشد. عبارت دوم ( $L_{conf}$ ) نیز خطای تشخیص کلاس به شیء داخل جعبه‌ی مرزی را ارزیابی می‌کند.

### ۳ - ۴ - گزارش نتایج

در مرحله پیاده‌سازی و آموزش شبکه اس‌اس‌دی از شبکه از پیش آموزش دیده شده‌ی VGG-16 [۶] برای استخراج ویژگی استفاده شده است. بهینه‌ساز مورد استفاده نزول گرادیان تصادفی<sup>۴۴</sup> بوده و تعداد نمونه آموزشی در هر دسته<sup>۴۵</sup> برابر با ۳۲ انتخاب شده است. نرخ یادگیری اولیه ۰.۰۰۱، تکانه<sup>۴۶</sup> ۰.۹ و کاهش وزن نیز تعیین گردیده است.

شبکه اس‌اس‌دی [۵] و روند آموزش آن برای پیش‌برد هر چه بهتر فرآیند تشخیص اشیاء و تعمیم‌پذیر ساختن مدل نهایی نسبت به تغییرات مکانی نظری چرخش<sup>۴۷</sup>، در زمان آموزش از افزونگی داده‌ها<sup>۴۸</sup> استفاده کرده و نمونه‌های تغییرات یافته برای آموزش به کار گرفته شده است.

<sup>43</sup> Grand Truth

<sup>44</sup> SGD (stastic Gradient Descent)

<sup>45</sup> Batch

<sup>46</sup> Momentum

<sup>47</sup> Rotation

<sup>48</sup> Data Augmentation

ارزیابی شبکه اس اس دی در دو نسخه و هر یک در دو حالت انجام و با شبکه های آرسی ان ان سریع<sup>۴۹</sup> [۹]، آرسی ان ان سریع تر<sup>۵۰</sup> [۱۰] و یولو [۴] مقایسه شده است که در جدول ۳-۱ قابل مشاهده است. مقصود از اس اس دی ۵۱۲ [۵] اضافه نمودن لایه های پیچشی بیشتر به معماری اصلی می باشد. در جدول ۳-۱ منظور از داده های ۷++۱۲۰ اجتماع مجموعه داده های PASCAL VOC [۷] نسخه های سال ۲۰۰۷ و ۲۰۱۲ میلاد می باشد که برای آموزش استفاده شده است و منظور از COCO ۷++۱۲۰ این است که آموزش شبکه با مجموعه داده های MS COCO [۱۱] انجام و تنظیم دقیق<sup>۵۱</sup> با مجموعه های ۷++۱۲۰ انجام شده است.

**جدول ۳-۱: مقایسه دقت شبکه اس اس دی با شبکه آرسی ان ان سریع، آرسی ان ان سریع تر و یولو [۵]**

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast[6]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster[2]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
Faster[2]	07++12+COCO	75.9	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2
YOLO[5]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD300	07++12+COCO	77.5	90.2	83.3	76.3	63.0	53.6	83.8	82.8	92.0	59.7	82.7	63.5	89.3	87.6	85.9	84.3	52.6	82.5	74.1	88.4	74.2
SSD512	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
SSD512	07++12+COCO	80.0	90.7	86.8	80.5	67.8	60.8	86.3	85.5	93.5	63.2	85.7	64.4	90.9	89.0	88.9	86.8	57.2	85.1	72.8	88.4	75.9

طبق نتایج حاصل ملاحظه می کنیم که شبکه اس اس دی ۵۱۲ بر اساس معیار ام ای پی توانسته است به بالاترین میانگین دقت در بین شبکه های آرسی ان ان سریع [۹]، آرسی ان ان سریع تر [۱۰] و یولو [۴] دست یابد که این نکته نشان می دهد قدرت مانور شبکه مذکور محدود به شبکه های تشخیص اشیای تک مرحله ای نبوده و بین سایر روش ها نیز موفق تر عمل کرده است. حال پرسشی که مطرح است عملکرد این شبکه به جهت سرعت چگونه است.

برای مقایسه زمان مورد نیاز برای تشخیص اشیاء بین شبکه های فوق، مجموعه داده های PASCAL VOC [۷] استفاده شده است؛ جدول ۲-۳ مقایسه مربوطه را نشان می دهد. معیار مقایسه تعداد فریم پردازشی در هر ثانیه می باشد که هر یک از شبکه ها توانسته اند فرآیند تشخیص را انجام دهد.

شبکه اس اس دی بر خلاف دقت خوبی که ارائه می کند، نتوانسته است در سرعت تشخیص عملکرد چندان مناسبی از خود نشان دهد. این شبکه [۵] توانسته است از شبکه های تشخیص اشیای مبتنی بر ناحیه های آرسی ان ان سریع [۹] و آرسی ان ان سریع تر [۱۰] در زمان سریع تر و با دقت بیشتری فرآیند تشخیص را انجام دهد اما در

<sup>49</sup> Fast R-CNN

<sup>50</sup> Faster R-CNN

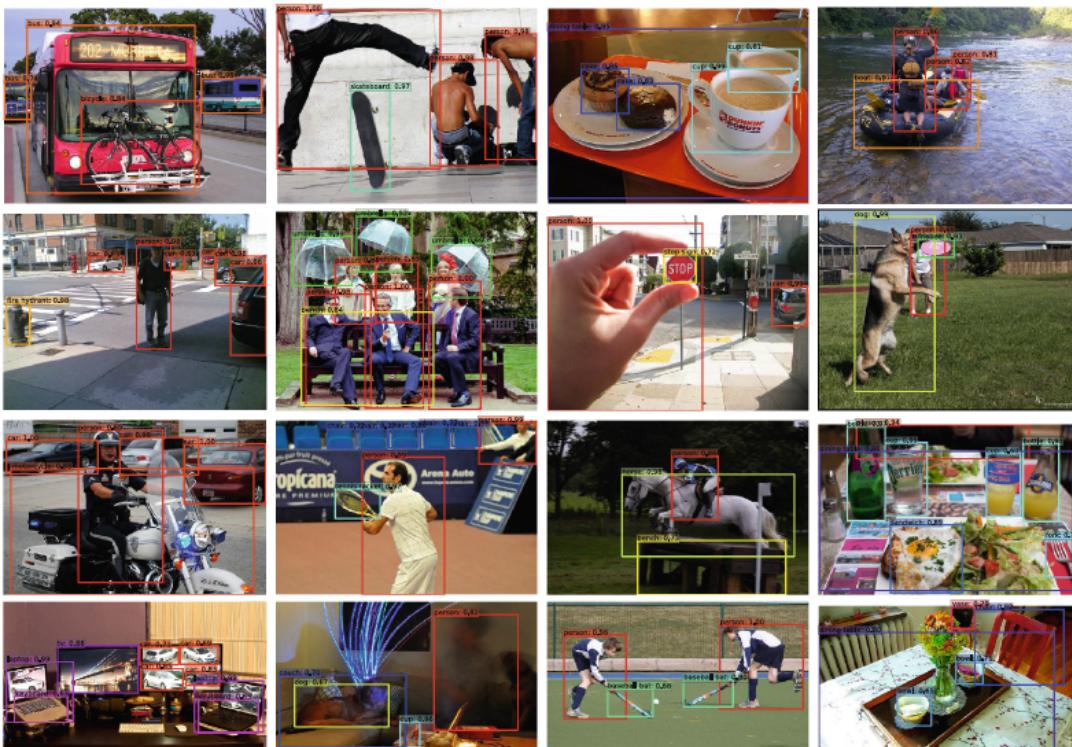
<sup>51</sup> Fine-Tune

مقابله با شبکه‌ی تشخیص اشیای تک مرحله‌ای یولو [۴] ناتوان بوده و با اینکه به دقت بیشتری دست یافته است ولی کندر از آن عمل کرده است و گزینه‌ی چندان مناسبی برای تشخیص اشیای بلاذرنگ نمی‌باشد.

جدول ۳-۲: مقایسه سرعت شبکه اس‌اس‌دی با شبکه آرسی‌ان‌ان سریع، آرسی‌ان‌ان سریع‌تر و یولو [۵]

Method	<i>mAP</i>	FPS	Test batch size	# Boxes
Faster R-CNN [2] (VGG16)	73.2	7	1	300
Faster R-CNN [2] (ZF)	62.1	17	1	300
YOLO [5]	63.4	45	1	98
Fast YOLO [5]	52.7	155	1	98
SSD300	74.3	46	1	8732
SSD512	76.8	19	1	24564
SSD300	74.3	59	8	8732
SSD512	76.8	22	8	24564

البته میزان تفاوت نیز قابل بیان است که شبکه اس‌اس‌دی نزدیک ۵۰٪ عملکردی بهتر نسبت به شبکه یولو داشته است اما نسبت به آن ۸۵٪ کندر عمل کرده است. شکل نمونه‌هایی از نتایج تشخیص اشیا توسط شبکه‌ی اس‌اس‌دی را نشان می‌دهد. قابل مشاهده است که قدرت و دقت این شبکه در تشخیص اشیای پیچیده و متداخل بالا می‌باشد و می‌تواند جعبه‌های مرزی را با دقت مناسبی تعیین نماید.



شکل ۳-۳: نمونه‌هایی از تشخیص اشیاء توسط شبکه اس‌اس‌دی [۵]

## **فصل چهارم اصلاح تابع هزینه**

#### ۴ - ۱ - شبکه‌ی تشخیص اشیای متراکم (رتینا)

در سال ۲۰۱۷ میلادی و پس از معرفی شبکه‌های تشخیص اشیای تک مرحله‌ای یولو [۴] و اس‌اس‌دی [۵]، پژوهش تشخیص اشیای متراکم (رتینا)<sup>۵۲</sup> [۱۲] با هدف بهبود دقت پیش‌بینی و کاهش خطای دسته‌بندی ارائه شده است که مشتمل بر معرفی تابع هزینه‌ای جدید برای آموزش شبکه و ساختاری جدید مبتنی بر شبکه اف‌پی‌ان [۳] می‌باشد. در بخش ۲-۴ به چالش موجود در توابع هزینه مرسوم پرداخته شده و در بخش ۳-۴ راه حل پیشنهادی مورد بررسی قرار گرفته و در نهایت در بخش ۴-۴ نتایج حاصل از پیشنهادهای ارائه شده ارزیابی و مقایسه خواهد شد.

#### ۴ - ۲ - مشکل موجود

تا زمان ارائه شبکه رتینا [۱۲] دقت تشخیص اشیاء در مدل‌های دو مرحله‌ای بسیار بالاتر از مدل‌های تک مرحله‌ای بوده و این ناشی از ارائه نواحی مطلوب و تنک<sup>۵۳</sup> برای دسته‌بندی می‌باشد. با وجود پایین بودن زمان پردازشی و ساده‌تر بودن روند محاسباتی در مدل‌های تک مرحله‌ای، همچنان دقت حاصل از پیش‌بینی در آن کمتر از مدل‌های دو مرحله‌ای می‌باشد. دلیل این مشکل عدم توازن<sup>۵۴</sup> کلاس‌های پس‌زمینه<sup>۵۵</sup> و پیش‌زمینه<sup>۵۶</sup> برای تعیین وجود شیء در حین محاسبه‌ی خطا در زمان بروزرسانی وزن‌های شبکه می‌باشد.

قابل انتظار است که در تصاویر و مجموعه داده‌های آموزشی نواحی زیادی متعلق به کلاس پس‌زمینه می‌باشد و شی‌ای در آن موجود نیست و از این رو، نسبت متعادلی بین تعداد آن و کلاس پیش‌زمینه وجود ندارد؛ لذا برای مثال اگر نسبت این دو کلاس را  $1:1000$  در نظر بگیریم، زمانی که مدل همه‌ی نواحی را پس‌زمینه تشخیص دهد، به خطای بسیار کمی نائل می‌شود چرا که سنجش یک ناحیه و خطای حاصل از پیش‌بینی<sup>۱</sup> اشتباه آن تاثیری در مقابل  $1:1000$  ناحیه نداشته و خطای مجموع  $1:1001$  ناحیه بسیار کم می‌شود و این باعث می‌شود با وجود کاهش خطای هزینه، دقت و صحت پیش‌بینی بالاتر نرود.

به جهت مقداری نیز بخواهیم بررسی کنیم، اگر  $1:100000$  نمونه آموزشی آسان هر یک خطای حداقلی ۱.۰ داشته باشند، خطای حاصل برابر با  $1:10000$  خواهد بود و اگر  $1:100$  نمونه آموزشی سخت هر یک خطای  $2.3$

<sup>52</sup> Retina

<sup>53</sup> Sparse

<sup>54</sup> Imbalance

<sup>55</sup> Back-ground

<sup>56</sup> Fore-ground

داشته باشد، خطای حاصل برابر با  $230$  خواهد بود و مدل برای کاهش خطای خود به سمتی حرکت و تمایل پیدا خواهد کرد که خطای نمونه‌های آموزشی آسان کاهش یابد [۱۲].

به عبارتی دیگر، شبکه در حین یادگیری تعداد خیلی زیادی نمونه و نواحی آسان را می‌آموزد و خطای کلی خود را با شبیب مناسبی کاهش می‌دهد اما در مقابله با نمونه‌های سخت که تعداد آنها نیز کمتر است، نمی‌تواند موفق عمل کند چرا که خطای حاصل از آن اندک می‌باشد و این باعث می‌شود مدل روی داده‌های آسان به همگرایی رسیده و دیگر تا پایان آموزش نتواند وزن خود را برای مواجه با نمونه‌های سخت به‌هنگام درآورد و این مشکلی است که در شبکه‌های تشخیص اشیای تک مرحله‌ای باعث پایین ماندن دقت می‌شود.

همچنین، این مشکل باعث مقاوم‌پذیر<sup>۵۷</sup> نشدن شبکه‌ی تشخیص شی‌در مقابل داده‌های دیده نشده<sup>۵۸</sup> می‌شود که یک مشکل اساسی در مبحث شبکه‌های عصبی می‌باشد.

در عمل و زمان تحلیل ملاحظه می‌شود که این مشکل بیشتر در تصاویری رخ می‌دهد که اشیای موجود در آن بصورت متراکم وجود داشته باشد و تصویر صرفا شامل چند شیء کلی نباشد [۱۲]. به عبارتی دیگر، نمونه‌های آموزشی سخت همان تصاویری هستند که اشیا بصورت متراکم در تصویر ظاهر شده باشند.

#### ۴ - ۳ - راه حل پیشنهادی

راه حل پیشنهادی برای برای مشکل موجود و تعمیم‌پذیر ساختن مدل به نمونه‌های سخت، اصلاح تابع هزینه و محاسبه خطا در زمان آموزش می‌باشد. اگر تابع هزینه آنتروپی متقابل<sup>۵۹</sup> را به عنوان محاسبه‌ی خطای یک مدل دسته‌بند دودوئی<sup>۶۰</sup> (که قابل بسط است) طبق رابطه ۱-۴ در نظر بگیریم، ملاحظه می‌کنیم که تفاوتی بین خطای نمونه‌های آموزشی سخت با تعداد کمتر و نمونه‌های آموزشی آسان با تعداد بالا قائل نیست.

$$\text{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases}$$

رابطه (۱-۴) : تابع هزینه آنتروپی متقابل

<sup>57</sup> Robustness

<sup>58</sup> Unseen

<sup>59</sup> Cross-Entropy loss

<sup>60</sup> Binary Classification

در رابطه ۴-۱ و ۴-۲ بترتیب احتمال رخداد کلاس یک و برچسب نمونه ورودی می‌باشد. حال اگر عبارت منفی لگاریتم را از هر دو جمله‌ی آنتروپی متقابل در رابطه ۴-۱ عامل‌گیری کنیم، می‌توانیم آن را بصورت رابطه ۴-۲ بازنویسی کنیم.

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

$$\text{CE}(p, y) = \text{CE}(p_t) = -\log(p_t)$$

رابطه (۴-۲) : بازنویسی تابع هزینه آنتروپی متقابل

حسنی که بازنویسی تابع هزینه آنتروپی متقابل طبق رابطه ۴-۲ دارد، این است که می‌توان به ازای کلاس نمونه آموزشی از خطا معادل با آن آگاه بود. با توجه به نکته‌ی مذکور، می‌توان با تعریف و اعمال یک ضریب نظیر  $a_t$  برای تابع هزینه آنتروپی متقابل به ازای نمونه‌های هر یک از کلاس‌ها، مقدار خطای حاصل از نمونه‌های آموزشی سخت را افزایش و مقدار خطای حاصل از نمونه‌های آموزشی آسان را کاهش داد و به یکدیگر نزدیک نمود تا بدین گونه به یادگیری نمونه‌های آموزشی سخت نیز توجه شود [۱۲]؛ رابطه‌ی اصلاح شده‌ی تابع هزینه آنتروپی در رابطه ۴-۳ قابل مشاهده می‌باشد.

$$\text{CE}(p_t) = -\alpha_t \log(p_t)$$

رابطه (۴-۳) : اصلاح تابع هزینه‌ی آنتروپی متقابل و وزن دار کردن آن نسبت به کلاس

با وجود بهبد و اصلاح تابع هزینه‌ی متقابل طبق رابطه ۴-۳ به ازای هر یک از کلاس‌ها، مشکلی که پایرجاست ثابت ماندن ضریب  $a_t$  نسبت به نمونه‌های یک کلاس می‌باشد که همانند یک ابرپارامتر<sup>۶۱</sup> رفتار می‌کند و این چالشی که دارد لحاظ نمودن اهمیت و تمایز خود هر یک از نمونه‌ها در زمان محاسبه‌ی خطا می‌باشد [۱۲].

برای حل این مشکل، پژوهشگران تابع خطای کانونی<sup>۶۲</sup> را طبق رابطه (را معرفی کرده‌اند [۱۲]) که در آن خطای حاصل از نمونه‌های آموزشی سخت و اشتباه دسته‌بندی شده<sup>۶۳</sup> تاثیر بیشتری داشته (با اعمال یک پارامتر گاما بصورت توانی) و وزن بیشتری نیز به خود می‌گیرند و این در حالی است که خطای حاصل از نمونه‌های آسان با دسته‌بندی صحیح تاثیر کمتری داشته و وزن کمتری نیز به خود می‌گیرند تا بدین گونه تعادلی بین خطاهای

<sup>61</sup> Hyper-Parameter

<sup>62</sup> Focal loss function

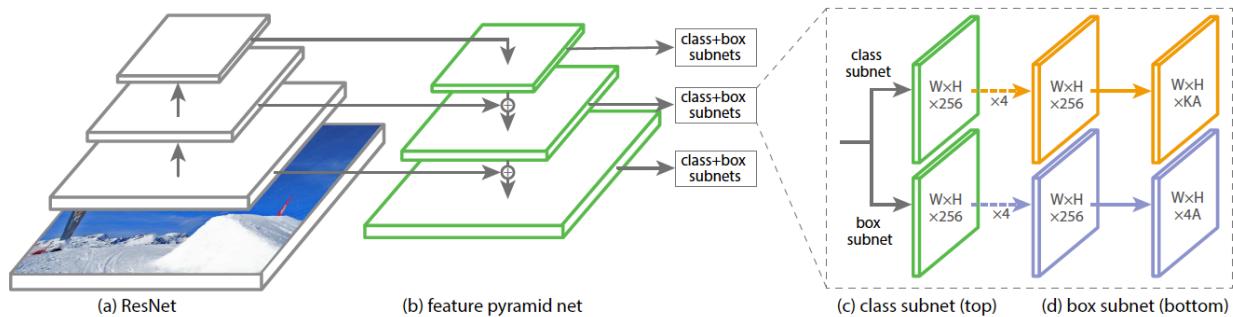
<sup>63</sup> Missclassified

حاصل ایجاد گردد. در بخش ۴-۴ نتایج حاصل از تعریف تابع هزینه آنتروپی کانونی مورد بررسی قرار گرفته و نشان داده خواهد شد که چگونه می‌توان میزان تاثیر را نیز کنترل نمود.

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

رابطه (۴-۴) : تابع هزینه کانونی

پس از معرفی تابع خطای کانونی، بایستی تاثیر آن را در آموزش یک شبکه‌ی تشخیص اشیای تک مرحله‌ای ارزیابی نمود؛ شبکه‌ی رتینا [۱۲] با این هدف معرفی شده است که معماری آن در شکل ۱-۴ قابل مشاهده است.



شکل ۱-۴: معماری شبکه رتینا [۱۲]

شبکه‌ی رتینا [۱۲] از ساختار هرمی<sup>۶۴</sup> شبکه اف پی ان [۳] به عنوان چارچوب اصلی خود استفاده کرده و از شبکه‌ی عصبی جلو-رو باقیمانده<sup>۶۵</sup> [۱۳] بهره گرفته است تا بتواند ویژگی‌های چند مقیاسی<sup>۶۶</sup> و قدرتمند هرمی را تولید نماید. خروجی هر سطح از هرم ویژگی به دو زیر شبکه‌ی<sup>۶۷</sup> مجزا از هم انتقال داده می‌شود؛ یکی از زیر شبکه‌ها مسئولیت دسته‌بندی را بر عهده داشته و دیگری جعبه‌ی مرزی بدست آمده را نسبت به حقیقت مبنا می‌سنجد.

<sup>64</sup> Pyramid

<sup>65</sup> Residual

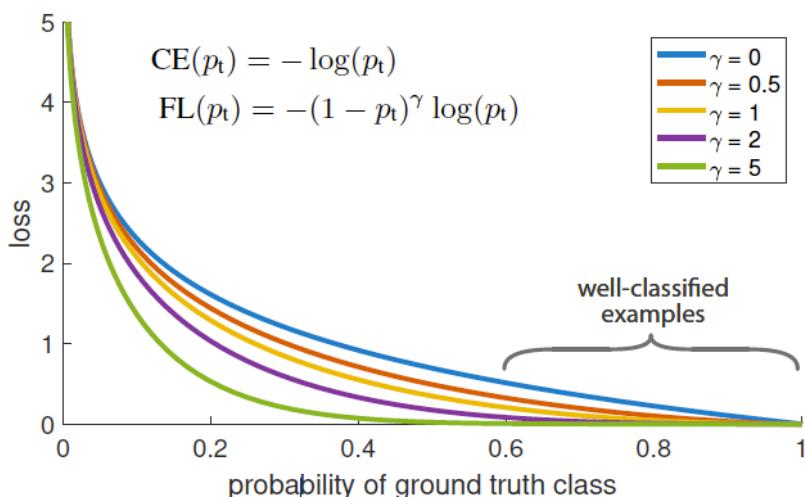
<sup>66</sup> Multi-scale

<sup>67</sup> Subnet

#### ۴ - ۴ - گزارش نتایج

ارزیابی تابع هزینه‌ی کانونی و شبکه‌ی رتینا [۱۲] را می‌توان در دو حوزه بیان و آن را تحلیل نمود. مبحث اول مربوط به میزان کاهش خطای استفاده از تابع هزینه کانونی نسبت به احتمال مثبت کلاس رخداد صحیح می‌باشد که در شکل ۲-۴ قابل مشاهده است.

تابع هزینه‌ی کانونی را می‌توان با ضرایب متفاوت اعمال نمود که مقایسه‌ی آن نشان می‌دهد که با افزایش ضریب گاما و تاثیر بیشتر دادن به نمونه‌های اشتباه دسته‌بندی شده، خطای مربوطه با شبیب بیشتر کاهش می‌یابد که این اثبات می‌کند اعمال و استفاده از تابع هزینه‌ی کانونی مناسب بوده و اثر مثبت بر آموزش شبکه دارد.



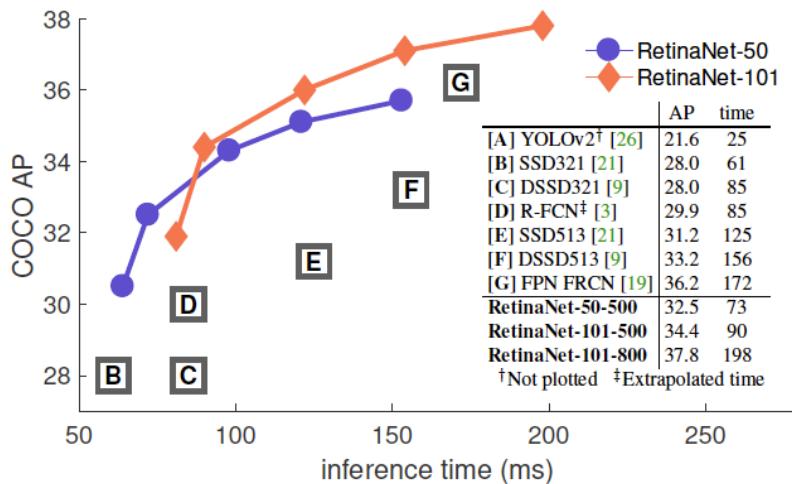
شکل ۲-۴: نمودار بررسی تابع هزینه کانونی با تغییرات ضریب تاثیر [۱۲]

مورد دوم برای گزارش مربوط به سرعت و دقت شبکه رتینا [۱۲] می‌باشد که با تابع هزینه کانونی آموزش دیده است. برای این منظور، از مجموعه داده MS COCO [۱۱] استفاده شده و میانگین دقت<sup>۶۸</sup> برای شبکه‌های مختلف توأم با زمان مورد نیاز برای خروجی نتیجه<sup>۶۹</sup> گزارش شده است که در شکل ۳-۴ قابل مشاهده می‌باشد.

<sup>۶۸</sup> AP (Average Precision)

<sup>۶۹</sup> Inference time

ملاحظه می‌کنیم که با اعمال تابع هزینه‌ی کانونی معرفی شده [۱۲] بر روی شبکه‌ی تشخیص اشیای تک مرحله‌ای متراکم [۱۲] علاوه بر سرعت، دقیق‌تر افزایش چشمگیر داشته است و در تمام مقایسه‌ها شامل مقایسه‌ی دقیق در زمان یکسان و مقایسه‌ی زمان در دقیق یکسان شبکه‌ی رتینا [۱۲] توانسته بهترین نتیجه را به ارمغان آورد.



شکل ۴-۳: مقایسه سرعت و دقیق‌تر شبکه رتینا [۱۲]

## **فصل پنجم تشخیص مرکز اشیاء**

تا سال ۲۰۱۹ میلادی، شبکه‌های تشخیص اشیای ارائه شده مبتنی بر استخراج و ارزیابی جعبه‌های مرزی پیشنهادی می‌باشد بطوریکه که شبکه‌های موجود نظیر یولو [۴] یا اس‌اس‌دی [۵] چند هزار جعبه‌ی مرزی ممکن را در تصویر ورودی بررسی نموده و پس از اعمال سرکوب غیرحداکثری‌ها اقدام به تعیین جعبه‌های مرزی نهایی می‌کند.

در بخش ۱-۵ مشکل موجود در استفاده از جعبه‌های مرزی پیشنهادی بیان شده و تاثیرات ناشی از آن در فرآیند تشخیص اشیاء مورد بررسی قرار می‌گیرد و در بخش ۲-۵ راهکار توسط شبکه‌ی تشخیص اشیای مرکز محور [۱۴] که هدف آن تشخیص مرکز اشیاء با کمک از گوشه‌های آن می‌باشد توضیح داده شده و در نهایت و در بخش ۳-۵ گزارش نتایج شبکه مذکور با شبکه‌های قبلی مقایسه خواهد شد.

### ۵ - ۱ - مشکل موجود در جعبه‌های مرزی پیش‌فرض

مشکلی که در شبکه‌های مرسوم تشخیص اشیاء نظیر یولو [۴]، اس‌اس‌دی [۵] و رتینا [۱۲] وجود دارد، تولید و پردازش نواحی بسیار زیاد بیهوده و زباله<sup>۷۰</sup> در حد چند هزار برای جستجوی اشیاء در تصویر ورودی می‌باشد. این فرآیند جستجو دو مشکل و نقص اساسی دارد که عبارت است از:

۱. پیچیدگی موجود در شبکه‌ها با مجذور تعداد پیش‌بینی‌ها متناسب است که این پیچیدگی در مدل‌های پیشنهادی با دقت، تعداد جعبه‌ی بالقوه<sup>۷۱</sup> بررسی شده و وضوح بالای تصویر ورودی ارتباط مستقیم داشته و با افزایش هر یک، پیچیدگی نیز افزایش می‌یابد [۱۵].

۲. شبکه‌های مذکور فوق برای تصمیم‌گیری نهایی در خصوص اشیای تشخیص داده شده بایستی منتظر نتایج تمام نواحی قابل بررسی و سرکوب غیرحداکثری باشد تا در نهایت بتواند جعبه‌های مرزی و اشیای داخل آن را پیش‌بینی کند. این انتظار باعث بالا رفتن زمان پردازشی و مصرف منابع می‌شود که در کابردهای بلادرنگ حوزه بینایی کامپیوتر قابل قبول نمی‌باشد [۱۵].

با توجه به دو مشکل مذکور، می‌توان جمع‌بندی نمود که بررسی تمامی چند هزار ناحیه‌ی بالقوه برای وجود یا عدم وجود شیء، بهینه نبوده و مشکل پیچیدگی و زمان را توامان دارد. برای حل این چالش، شبکه‌های

<sup>70</sup> Garbage

<sup>71</sup> Potenstial

تشخیص اشیای گوشه محور [۱۵] و مرکز محور [۷۲] پیشنهاد شده اند که معماری و ساختاری پیشنهادی آن در ادامه مورد اشاره قرار خواهد گرفت.

### ۵ - ۲ - راهکار شبکه مرکز محور

معماری شبکه مرکز محور [۱۶] با الهام از روش‌های سنتی<sup>۷۴</sup> و مبتنی بر نقاط کلیدی<sup>۷۵</sup> برای حل مشکلات موجود مذکور در بخش ۲-۵ پیشنهاد شده است. روش ارائه شده بر اساس تولید نقشه حرارت<sup>۷۶</sup> وجود اشیا و تخمین گوشه‌های جعبه‌ی متناظر بنا شده است. در زمان آموزش بدین صورت عمل می‌شود که مرکز جعبه‌ی حقیقت مبنا به عنوان مرکز شیء به عنوان یک قله<sup>۷۷</sup> در نظر گرفته شده و سپس با یک فیلتر گوسی<sup>۷۸</sup> هموار می‌شود.

مدل شبکه مرکز محور [۱۶] که معماری آن نیز در شکل ۱-۵ قابل مشاهده است، از دو بخش مجزا از هم برای تشخیص اشیاء و پیش‌بینی استفاده می‌کند. بخش اول اقدام به تولید و تخمین نقشه‌ی دو بعدی حرارت می‌کند که نشان می‌دهد اشیاء با چه احتمالی در هر نقطه حضور دارند و نقشه‌ی مذکور در صدد تخمین حقیقت مبنا می‌باشد. بخش دوم در قالب مسئله‌ی رگرسیونی اقدام به تخمین جعبه‌ی مرزی برای اشیاء می‌کند که این تخمین در قالب دو نقطه ارائه می‌شود. از دو نقطه‌ی مذکور یکی نقطه‌ی چپ-بالا و دیگری نقطه‌ی راست-پایین جعبه‌ی مرزی می‌باشد.

در نهایت خروجی دو بخش فوق با یکدیگر ترکیب شده و پیش‌بینی نهایی را برای تشخیص اشیاء حاصل می‌کند که در این صورت دیگر نیازی به اعمال سرکوب غیرحداکثری و انتظار برای پردازش‌های مرتبط نمی‌باشد. فرآیند ترکیب نتایج بصورت جزئی‌تر و در قالب گام‌های زیر انجام می‌پذیرد:

۱. روی نقشه‌ی حرارت بدست آمده، فیلتر بیشنه رای<sup>۷۹</sup> اعمال می‌شود که باعث می‌شود مقادیر اکثر پیکسل‌ها افزایش یافته و در حول نقطه‌ی بیشینه یک سطح هموار پدید آید.

<sup>72</sup> Corner

<sup>73</sup> Center

<sup>74</sup> Traditional

<sup>75</sup> Keypoints

<sup>76</sup> Heat Map

<sup>77</sup> Peak

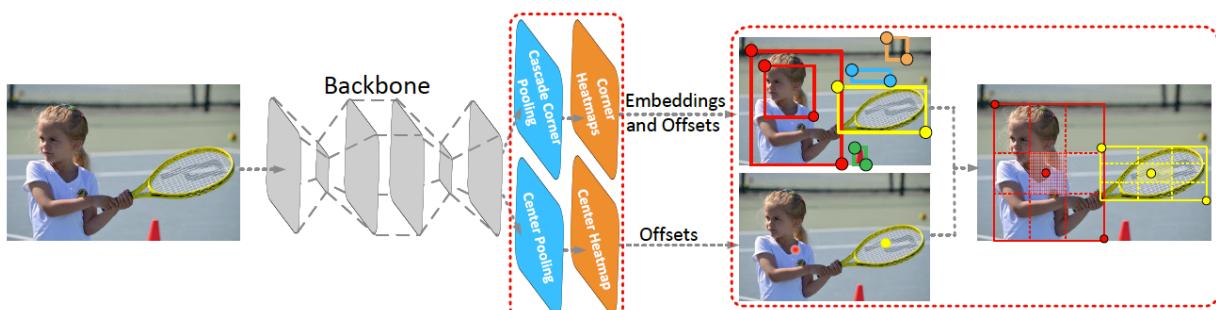
<sup>78</sup> Gaussian Filter

<sup>79</sup> Max pooling

۲. حال بصورت عملیات بولی<sup>۸۰</sup> هر یک از پیکسل های نقشهی حرارت با پیکسل متناظر خود در خروجی قسمت قبل را مقایسه می شود؛ در صورتی که مقادیر هر یک از پیکسل ها عوض شده باشد، مقدار صفر و در غیر اینصورت مقدار یک قرار داده می شود. در این مرحله یک ماسک<sup>۸۱</sup> تولید می شود که نشان دهنده جایگاه نقاط بیشینه در نقشه حرارتی است.

۳. ماسک به وجود آمده در قسمت قبل به نقشهی حرارتی ضرب می شود تا صرفا نقاطی از نقشهی حرارتی باقی بماند که بیشینه محلی بوده و احتمال وجود اشیاء به مرکزیت آن بالا است. حال در این نقشه، با اعمال یک حد آستانه نقاطی از تصویر بدست می آید که برای مرکز بودن یک شیء محتمل تر از بقیه نقاط می باشد.

۴. حال نقاط مرکزی بدست آمده با جعبه حاصل در بخش دوم مدل تطبیق داده می شود؛ در صورتی که نقاط مرکزی اشیاء در داخل جعبه های مرزی متناظر با خود قرار گرفته باشد، جعبهی مرزی بدست آمده به عنوان جعبهی حاوی شیء شناسایی و خروجی داده می شود.



شکل ۵: معماری شبکه تشخیص اشیای مرکز محور [۱۴]

توضیحات فوق مربوط عملکرد و معماری شبکه مرکز محور [۱۴] بوده و تابع هزینهی آن طبق رابطه ۵-۱ محاسبه می شود. در این تابع بالنویس<sup>۸۲</sup> های ce و co بترتیب مربوط به خطای نقاط مرکزی و نقاط گوشه های

<sup>80</sup> Boolean

<sup>81</sup> Mask

<sup>82</sup> Superscript

بدست آمده می‌باشد. پایین نویس<sup>۸۳</sup> های  $\text{det}$  و  $\text{off}$  مربوط به خطای شناسایی و خطای فاصله‌ی حاشیه‌ای<sup>۸۴</sup> می‌باشد. همچنین پایین نویس‌های  $\text{pull}$  و  $\text{push}$  بترتیب خطای کمینه‌سازی<sup>۸۵</sup> و بیشینه‌سازی<sup>۸۶</sup> فاصله‌ی نقطه‌ی مرکزی اشیای مشابه با نقاط گوشه‌ای خود می‌باشد تا نقاط مرکزی اشیای مختلف در داخل نقاط گوشه‌ای اشیای دیگر شناسایی نشود.

$$L = L_{\text{det}}^{\text{co}} + L_{\text{det}}^{\text{ce}} + \alpha L_{\text{pull}}^{\text{co}} + \beta L_{\text{push}}^{\text{co}} + \gamma (L_{\text{off}}^{\text{co}} + L_{\text{off}}^{\text{ce}})$$

رابطه (۱-۵) :تابع هزینه‌ی شبکه تشخیص اشیای مرکز محور

همچنین تابع هزینه‌ی مورد استفاده در شبکه‌ی تشخیص اشیای مرکز محور، تابع هزینه‌ی کانونی [۱۲] می‌باشد تا از نامتعادل بودن یادگیری جلوگیری بعمل آید.

### ۵ - ۳- گزارش نتایج

نتایج مقایسه‌ی دقت شبکه تشخیص اشیای مرکز محور [۱۴] با سایر شبکه‌ها در جدول ۱-۵ قابل مشاهده است. این جدول شامل مقایسه با شبکه‌های دو و تک مرحله‌ای می‌باشد که بروی مجموعه داده‌ی MS COCO [۱۱] ارزیابی شده است. قابل ملاحظه است که شبکه مرکز محور [۱۱] در بین شبکه‌های تشخیص اشیای تک مرحله‌ای بهترین نتایج را بدست آورده است.

مورد قابل ذکر بعدی برأساس نتایج حاصل، اختلاف بسیار اندک شبکه‌ی مذکور با شبکه‌های دو مرحله‌ای تشخیص اشیاء می‌باشد که نشان می‌دهد دقت و صحت پیش‌بینی بهم نزدیک‌تر شده و حتی در برخی موارد پیشی گرفته است که حاکی از موفقیت آمیز بودن ایده‌ی ارائه شده [۱۴] است.

نکته‌ی مورد ابهام موجود، مقایسه‌ی زمان اجرا و پیش‌بینی شبکه‌ی مرکز محور [۱۴] با سایر شبکه‌ها می‌باشد که به آن اشاره نشده است و صرفا با شبکه گوشه محور [۱۵] مقایسه انجام و بر حسب میلی ثانیه گزارش شده است که در بهترین حالت ۳۰ میلی ثانیه عملکرد سریع‌تر و در بدترین حالت ۴۰ میلی ثانیه عملکرد کندر داشته است که بسته به شبکه‌ی پایه‌ی انتخابی تغییر داشته است.

<sup>83</sup> Subscript

<sup>84</sup> Offset

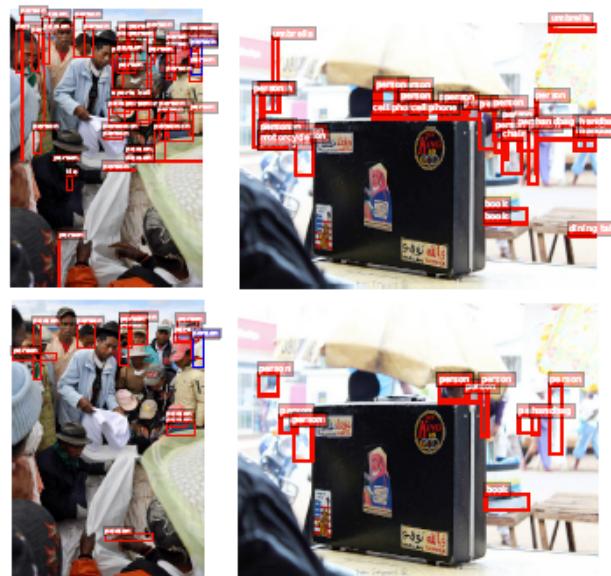
<sup>85</sup> Minimization

<sup>86</sup> Maximization

جدول ۱-۵: جدول مقایسه‌ی شبکه‌ی تشخیص اشیای مرکز محور با سایر شبکه‌ها

Method	Backbone	Train input	Test input	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>	AR <sub>1</sub>	AR <sub>10</sub>	AR <sub>100</sub>	AR <sub>S</sub>	AR <sub>M</sub>	AR <sub>L</sub>
<b>Two-stage:</b>															
DeNet [43]	ResNet-101 [15]	512×512	512×512	33.8	53.4	36.1	12.3	36.1	50.8	29.6	42.6	43.5	19.2	46.9	64.3
CoupleNet [52]	ResNet-101 ori.	ori.	ori.	34.4	54.8	37.2	13.4	38.1	50.8	30.0	45.0	46.4	20.7	53.1	68.5
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [42]	~ 1000×600	~ 1000×600	34.7	55.5	36.7	13.5	38.1	52.0	-	-	-	-	-	-
Faster R-CNN +++ [15]	ResNet-101	~ 1000×600	~ 1000×600	34.9	55.7	37.4	15.6	38.7	50.9	-	-	-	-	-	-
Faster R-CNN w/ FPN [24]	ResNet-101	~ 1000×600	~ 1000×600	36.2	59.1	39.0	18.2	39.0	48.2	-	-	-	-	-	-
Faster R-CNN w/ TDM [38]	Inception-ResNet-v2	-	-	36.8	57.7	39.2	16.2	39.8	52.1	31.6	49.3	51.9	28.1	56.6	71.1
D-FCN [7]	Aligned-Inception-ResNet	~ 1000×600	~ 1000×600	37.5	58.0	-	19.4	40.1	52.5	-	-	-	-	-	-
Regionlets [46]	ResNet-101	~ 1000×600	~ 1000×600	39.3	59.8	-	21.7	43.7	50.9	-	-	-	-	-	-
Soft-NMS [2]	Aligned-Inception-ResNet	~ 1300×800	~ 1300×800	40.9	62.8	-	23.3	43.6	53.3	-	-	-	-	-	-
Fitness R-CNN [44]	ResNet-101	512×512	1024×1024	41.8	60.9	44.9	21.5	45.0	57.5	-	-	-	-	-	-
Grid R-CNN w/ FPN [29]	ResNeXt-101	~ 1300×800	~ 1300×800	43.2	63.0	46.6	25.1	46.5	55.2	-	-	-	-	-	-
D-RFCN + SNIP (multi-scale) [39]	DPN-98 [5]	~ 2000×1200	~ 2000×1200	45.7	67.3	51.1	29.3	48.8	57.1	-	-	-	-	-	-
PANet (multi-scale) [27]	ResNeXt-101	~ 1400×840	~ 1400×840	47.4	67.2	51.8	30.1	51.7	60.0	-	-	-	-	-	-
<b>One-stage:</b>															
YOLOv2 [33]	DarkNet-19	544×544	544×544	21.6	44.0	19.2	5.0	22.4	35.5	20.7	31.6	33.3	9.8	36.5	54.4
DSOD300 [35]	DS/64-192-48-1	300×300	300×300	29.3	47.3	30.6	9.4	31.5	47.0	27.3	40.7	43.0	16.7	47.1	65.0
GRP-DSOD320 [36]	DS/64-192-48-1	320×320	320×320	30.0	47.9	31.8	10.9	33.6	46.3	28.0	42.1	44.5	18.8	49.1	65.0
SSD513 [28]	ResNet-101	513×513	513×513	31.2	50.4	33.3	10.2	34.5	49.8	28.3	42.1	44.4	17.6	49.2	65.8
DSSD513 [9]	ResNet-101	513×513	513×513	33.2	53.3	35.2	13.0	35.4	51.1	28.9	43.5	46.2	21.8	49.1	66.4
RefineDet512 (single-scale) [48]	ResNet-101	512×512	512×512	36.4	57.5	39.5	16.6	39.9	51.4	-	-	-	-	-	-
CornerNet511 (single-scale) [21]	Hourglass-52	511×511	ori.	37.8	53.7	40.1	17.0	39.0	50.5	33.9	52.3	57.0	35.0	59.3	74.7
RetinaNet800 [25]	ResNet-101	800×800	800×800	39.1	59.1	42.3	21.8	42.7	50.2	-	-	-	-	-	-
CornerNet511 (multi-scale) [21]	Hourglass-52	511×511	≤1.5×	39.4	54.9	42.3	18.9	41.2	52.7	35.0	53.5	57.7	36.1	60.1	75.1
CornerNet511 (single-scale) [21]	Hourglass-104	511×511	ori.	40.5	56.5	43.1	19.4	42.7	53.9	35.3	54.3	59.1	37.4	61.9	76.9
RefineDet512 (multi-scale) [48]	ResNet-101	512×512	≤2.25×	41.8	62.9	45.7	25.6	45.1	54.1	-	-	-	-	-	-
CornerNet511 (multi-scale) [21]	Hourglass-104	511×511	≤1.5×	42.1	57.8	45.3	20.8	44.8	56.7	36.4	55.7	60.0	38.5	62.7	77.4
CenterNet511 (single-scale)	Hourglass-52	511×511	ori.	41.6	59.4	44.2	22.5	43.1	54.1	34.8	55.7	60.1	38.6	63.3	76.9
CenterNet511 (single-scale)	HRNet-W64 [41]	511×511	ori.	44.0	62.6	47.1	23.0	47.3	57.8	35.4	56.9	61.7	38.3	66.2	79.6
CenterNet511 (single-scale)	Hourglass-104	511×511	ori.	44.9	62.4	48.1	25.6	47.4	57.4	36.1	58.4	63.3	41.3	67.1	80.2
CenterNet511 (multi-scale)	Hourglass-52	511×511	≤1.8×	43.5	61.3	46.7	25.3	45.3	55.0	36.0	57.2	61.3	41.4	64.0	76.3
CenterNet511 (multi-scale)	HRNet-W64	511×511	≤1.8×	46.3	64.7	49.8	26.6	49.6	59.3	36.8	58.6	62.9	42.1	66.9	79.0
CenterNet511 (multi-scale)	Hourglass-104	511×511	≤1.8×	47.0	64.5	50.7	28.9	49.9	58.9	37.5	60.3	64.8	45.1	68.3	79.7

شکل ۲-۵ حاصل برخی نتایج بصری تشخیص اشیای شبکه‌ی مرکز محور [۱۴] می‌باشد که نشان می‌دهد خطای جعبه‌های مرزی پیش‌بینی شده‌ی بسیار کوچه کاهش یافته است.



شکل ۵-۲: بهبود تشخیص اشیای بسیار کوچک در شبکه‌ی مرکز محور [۱۴]

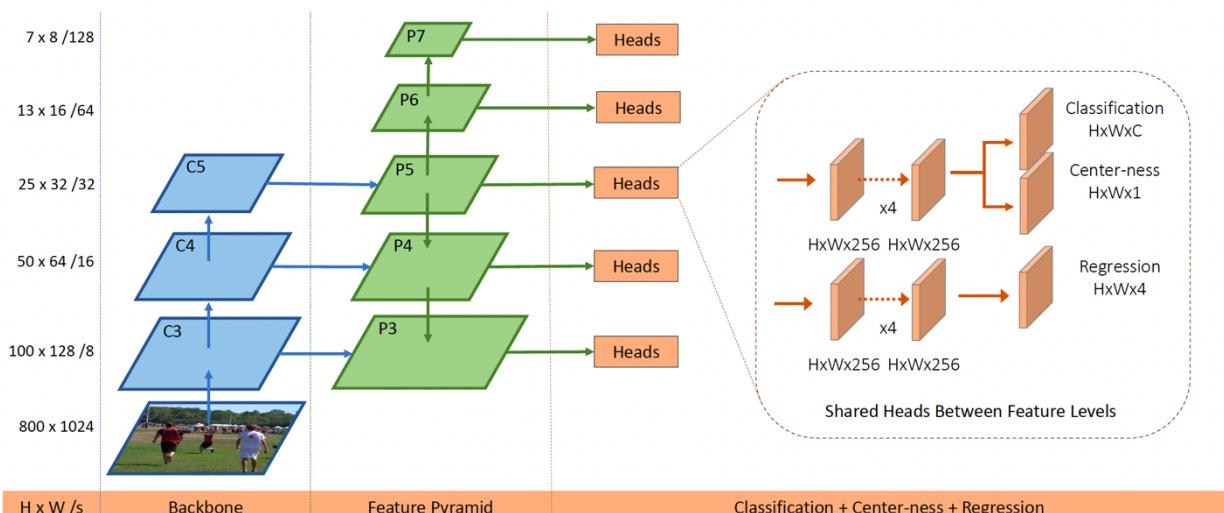
## **فصل ششم استفاده از پیکسل**

شبکه‌های تشخیص اشیای تک مرحله‌ای بررسی شده تا کنون، همگی جعبه‌های مرزی پیشنهادی و از قبل مشخص شده در مقیاس‌های مختلف که تعداد آن گاهای به چندین هزار مورد رسیده است را برای حضور و تشخیص شیء به کار برد و در نهایت با ترکیب نتایج و پردازش‌هایی نظیر محاسبه نسبت اشتراک به اجتماع جعبه‌ها اقدام به پیش‌بینی کرده‌اند که باز محاسباتی قابل ملاحظه‌ای را نیز به همراه داشته‌اند. نکته‌ی بعدی قابل اشاره تعداد ابرپارامترهای مورد نیاز در آموزش شبکه‌های مذکور می‌باشد که باعث می‌شود نتایج حاصل در تکرار آزمایش‌های مختلف تا چند درصد از هم متفاوت باشد.

شبکه تشخیص شیء کاملاً پیچشی [۱۶] با هدف مستقل ساختن شبکه‌های تشخیص اشیا از جعبه‌های مرزی و نواحی پیشنهادی<sup>۸۷</sup> و حل چالش‌های مذکور ارائه شده است که در این فصل به آن پرداخته خواهد شد. ابتدا در بخش ۱-۶ معرفی شده و سپس در بخش ۲-۶ تابع هزینه و آموزش شبکه موردن بررسی قرار گرفته و در بخش ۳-۶ گزارش نتایج حاصل از آن نشان داده شده و در نهایت مزیت‌های آن بیان خواهد شد.

## ۶ - ۱ - شبکه تشخیص شیء کاملاً پیچشی

طرحواره معماري شبکه تشخیص شیء کاملاً پیچشی [۱۶] به عنوان اولین شبکه‌ای که از تمام پیکسل‌های تصویر بجای جعبه‌های مرزی پیش‌فرض استفاده می‌نماید در شکل ۱-۶ قابل ملاحظه است.



شکل ۶-۱: ساختار شبکه تشخیص شیء کاملاً پیچشی [۱۶]

<sup>87</sup> Proposal Region

در شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶] از شبکه‌ی ResNet-50 [۱۳] به عنوان ستون استخراج ویژگی<sup>۸۸</sup> استفاده شده است. ابعاد ورودی تصویر به شبکه در آزمایش‌های انجام شده ۱۰۲۴\*۸۰۰ پیکسل بوده و خروجی نیز بردارهایی هم اندازه با تصویر ورودی می‌باشد. از هر تصویر ورودی صفحات ویژگی در سه مقیاس استخراج شده و سپس با گذر از یک لایه پیچشی به هرم ویژگی [۳] داده شده و پس از حاصل شدن صفحات ویژگی‌ها در ۵ مقیاس، هر یک بصورت مستقل به سر پیش‌بینی<sup>۸۹</sup> مشترک داده می‌شود تا بردارهای هم اندازه‌ی تصویر حاصل شود.

ایده ارائه شده در شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶] مبتنی بر دسته‌بندی و پیش‌بینی هر یک از پیکسل‌های تصویر می‌باشد. در طی شبکه به ازای هر پیکسل موارد زیر پیش‌بینی می‌شود:

۱. اگر تعیین یک جعبه‌ی مرزی توسط یک پیکسل را بخواهیم نمایش دهیم، به چهار مقدار برای فاصله از چهار جهت اصلی (بالا، پایین، چپ و راست) نیازمندیم که موارد یاد شده به ازای هر پیکسل توسط مدل رگرسیونی تخمین<sup>۹۰</sup> زده می‌شود. مثال این مورد در شکل ۲-۶ قابل مشاهده است.

۲. به ازای جعبه‌ی مرزی مشخص شده متناظر با هر پیکسل، امتیاز<sup>۹۱</sup> و احتمال وجود هر یک از کلاس‌های آموزشی در آن جعبه‌ی مرزی پیش‌بینی می‌شود که به تعداد کلاس‌های مجموعه داده وابسته می‌باشد که برای مجموعه داده MS COCO [۱۱] برابر با ۸۰ می‌باشد. در بخش بعد چگونگی اختصاص امتیاز به هر کلاس مورد بررسی قرار خواهد گرفت.

۳. در نهایت به ازای هر پیکسل فاصله‌ی مرکزیت<sup>۹۲</sup> محاسبه می‌شود. این معیار فاصله‌ی پیکسل را از مرکز جعبه‌ی مرزی نشان می‌دهد. این معیار به تابع هزینه‌ی شبکه افزوده شده و در زمان آزمون نیز این مقدار در امتیاز بردارهای حاصل ضرب می‌شود تا بصورت غیرضمونی سرکوب غیرحداکثری بین پیکسل‌ها اعمال شده و پیکسل‌هایی در مرکزیت هستند مقدار بیشتر داشته باشند.

<sup>88</sup> Backbone

<sup>89</sup> Prediction head

<sup>90</sup> Estimate

<sup>91</sup> Score

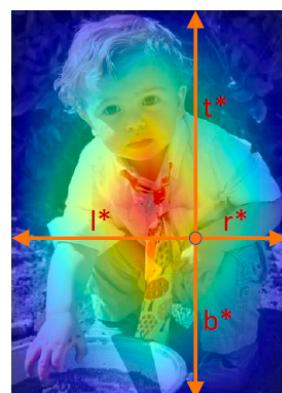
<sup>92</sup> Center-ness

مثال توضیحات فوق در شکل ۳-۶ قابل ملاحظه می‌باشد. اعمال معیار مذکور علاوه بر منطقی ساختن وضعیت هر یک از پیکسل‌ها، باعث بهبود عملکرد شده و دقت پیش‌بینی تشخیص شیء را افزایش داده است زیرا از ایجاد جعبه‌های مرزی مختلف در مقیاس‌های پایین‌تر جلوگیری کرده و خطا را کاهش می‌دهد.



شکل ۳-۶: چگونگی اختصاص یک جعبه‌ی مرزی به یک پیکسل در شبکه تشخیص شیء کاملاً پیچشی [۱۶]

بر اساس توضیحات فوق می‌توان هر پیکسل را در دسته‌ی مثبت<sup>۹۳</sup> یا دسته‌ی منفی<sup>۹۴</sup> قرار داد بطوری که اگر پیکسلی در داخل هر یک از حقیقت‌های مبنا قرار گرفته باشد به دسته‌ی مثبت اختصاص یافته و نمونه‌ی مثبت تلقی خواهد شد و در روند پیش‌بینی جعبه‌های مرزی حضور خواهد داشت و در صورتی که پیکسلی در داخل هیچ حقیقت مبنایی قرار نگرفته باشد، به دسته‌ی منفی قرار گرفته و با اختصاص کلاس صفر به عنوان پیکسل پس‌زمینه تلقی خواهد شد.



شکل ۳-۷: نتیجه‌ی اعمال معیار مرکزیت به ازای پیکسل‌ها در شبکه تشخیص شیء کاملاً پیچشی [۱۶]

<sup>۹۳</sup> Positive

<sup>۹۴</sup> Negative

یکی از چالش های موجود در شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶] رفع ابهام<sup>۹۵</sup> در پیکسل‌های می‌باشد که در داخل دو یا چند حقیقت مبنا قرار می‌گیرد و میتواند هر چند جعبه را در حاصل از مقیاس‌های مختلف تعیین کنند. تیان<sup>۹۶</sup> و همکاران [۱۶] بصورت تجربی و طی آزمایش‌هایی تجربی نشان داده‌اند که از جایی که پیش‌بینی هر یک از پیکسل‌ها در چند مقیاس بصورت مستقل از هم و با یک سر بیش‌بینی اشتراکی تعیین می‌شود؛ هر یک از پیکسل‌ها، جعبه‌ی مرزی متناظر با شیء حاضر را خروجی میدهند و به ندرت پیش‌بین می‌آید که ابهام در بین پیش‌بینی‌های مختلف یک پیکسل در مقیاس‌های وجود داشته باشد. طبق پیشنهاد تیان و همکاران [۱۶] در صورت وجود ابهام نیز کوچکترین جعبه‌ی مرزی پیشنهادی انتخاب می‌شود که نشان داده می‌شود این انتخاب عملکرد مناسبی را نیز ارائه می‌کند.

## ۶ - ۲ - تابع هزینه و آموزش

آموزش و ارزیابی شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶] با مجموعه داده‌ی MS COCO [۱۱] انجام پذیرفته و از از شبکه‌ی ResNet-50 [۱۳] به عنوان ستون استخراج ویژگی از تصویر استفاده شده است. امتیاز دهی به هر پیکسل به ازای هر کلاس آموزشی توسط مدل چند کلاسه<sup>۹۷</sup> انجام نپذیرفته و به ازای هر کلاس آموزشی یک مدل دودوئی آموزش دیده می‌شود و مدل متناظر امتیاز هر کلاس را تعیین می‌کند.

رابطه‌ی ۱-۶ تابع هزینه‌ی شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶] را نشان می‌دهد؛ در رابطه‌ی مذکور عبارت‌های بالاتریس با \* مقادیر حاصل از مقادیر حقیقت مبنا و برچسب را نشان داده و  $N_{\text{pos}}$  نیز نشان دهنده‌ی تعداد پیکسل‌های مثبت در جعبه‌ی مرزی پیشنهادی توسط پیکسل می‌باشد.

عبارت  $t$  نماینده‌ی ابعاد جعبه‌ی مرزی پیشنهادی توسط پیکسل و  $p$  نیز نماینده‌ی امتیازهای هر یک از کلاس‌های آموزشی می‌باشد. از دو عبارت موجود در رابطه، اولی خطای حاصل از دسته‌بندی پیکسل به کلاس‌های آموزشی را نشان داده و دومی نیز خطای حاصل از ابعاد جعبه‌ی مرزی را ایجاد می‌کند. در زمان آموزش، به تابع هزینه مقدار معیار مرکزیت پیکسل افزوده می‌شود در حالی که در زمان آزمون این مقدار در امتیاز کلاس‌های پیش‌بینی شده ضرب می‌شود تا وزن پیکسل‌های مرکزی بیشتر شود.

<sup>95</sup> Ambiguous

<sup>96</sup> Tian

<sup>97</sup> Multi-class

$$\begin{aligned} L(\{\mathbf{p}_{x,y}\}, \{\mathbf{t}_{x,y}\}) = & \frac{1}{N_{\text{pos}}} \sum_{x,y} L_{\text{cls}}(\mathbf{p}_{x,y}, c_{x,y}^*) \\ & + \frac{\lambda}{N_{\text{pos}}} \sum_{x,y} \mathbb{1}_{\{c_{x,y}^* > 0\}} L_{\text{reg}}(\mathbf{t}_{x,y}, t_{x,y}^*), \end{aligned}$$

رابطه (۱-۶) :تابع هزینه شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶]

### ۶ - ۳ - گزارش نتایج

نتایج و عملکرد شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶] در جدول ۱-۶ آورده شده است؛ همانطور که ملاحظه می‌شود شبکه مذکور با حذف جعبه‌ها و نواحی مرزی پیشنهادی پیش‌فرض چندین هزار تایی و ارائه روشی جدید مبتنی بر تخمین جعبه‌ی مرزی به ازای هر پیکسل توانسته است نتایج قبلی توسط شبکه‌های مختلف تشخیص اشیای تک و دو مرحله‌ای را بهبود دهد.

علاوه بر عملکرد و دقت بالای شبکه‌ی مذکور، بار پردازشی و تعداد پارامترهای مورد نیاز آن نیز کمتر از روش‌های دیگر می‌باشد چرا که چندین هزار نواحی و جعبه‌ی از قبل تعیین شده را بررسی نمی‌کند و درگیر چالش‌های ترکیب نواحی نیز نمی‌شود. مزیت بعدی این شبکه در تعمیم آن به فعالیت‌های دیگر نظیر قطعه‌بندی معنایی<sup>۹۸</sup> می‌باشد.

جدول ۱-۶ : مقایسه عملکرد شبکه‌ی تشخیص شیء کاملاً پیچشی با سایر شبکه‌های تک و دو مرحله‌ای [۱۶]

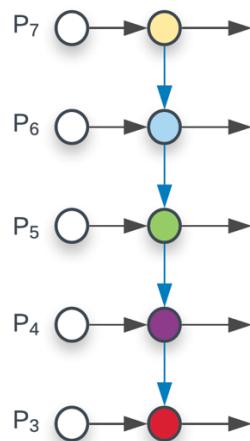
Method	Backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
Two-stage methods:							
Faster R-CNN w/ FPN [14]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [11]	Inception-ResNet-v2 [27]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w/ TDM [25]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	52.1
One-stage methods:							
YOLOv2 [22]	DarkNet-19 [22]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [18]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [5]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [15]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
CornerNet [13]	Hourglass-104	40.5	56.5	43.1	19.4	42.7	53.9
FSAF [34]	ResNeXt-64x4d-101-FPN	42.9	63.8	46.3	26.6	46.2	52.7
FCOS	ResNet-101-FPN	41.5	60.7	45.0	24.4	44.8	51.6
FCOS	HRNet-W32-51 [26]	42.0	60.4	45.3	25.4	45.0	51.0
FCOS	ResNeXt-32x8d-101-FPN	42.7	62.2	46.1	26.0	45.6	52.6
FCOS	ResNeXt-64x4d-101-FPN	43.2	62.8	46.6	26.5	46.2	53.3
FCOS w/ improvements	ResNeXt-64x4d-101-FPN	<b>44.7</b>	<b>64.1</b>	<b>48.4</b>	<b>27.6</b>	<b>47.5</b>	<b>55.6</b>

<sup>98</sup> Semantic segmentation

**فصل هفتم: بهبود ارتباطات هرم ویژگی در تشخیص اشیاء**

## ۷ - ۱ - چالش موجود در هرم ویژگی

استفاده از هرم ویژگی در مباحث بینایی کامپیوتر و فعالیتهای مختلف آن نظری تشخیص یا شناسایی اشیاء گذشته طولانی داشته و در استخراج ویژگی از تصویر به کار گرفته می‌شود. در شبکه‌های تشخیص شیء مبتنی بر هرم ویژگی [۳] اطلاعات و ویژگی‌های هر مقیاس از تصویر با مقیاس بعدی بصورت متوالی تاثیر و ارتباط دارد که طرحواره آن در شکل ۱-۷ قابل ملاحظه است.



شکل ۱-۷: طرحواره هرم ویژگی [۳]

مشکلی که در استفاده از هرم ویژگی بصورت فوق و مرسوم دارد این است که اطلاعات و ویژگی‌های حاصل در مقیاس‌های مختلف صرفاً یکطرفه با هم در ارتباط می‌باشد که باعث محدودیت در انتقال می‌شود. در پژوهش تان<sup>۹۹</sup> و همکاران [۱۷] ارائه یک لایه‌ی جدید هرم ویژگی دو طرفه مورد مطالعه قرار گرفته است که محدودیت مذکور را نداشته و اطلاعات بصورت دو طرفه در مقیاس‌های مختلف می‌تواند گذر داشته باشد.

همچنین در لایه‌ی ارائه شده در پژوهش تان و همکاران [۱۷] امکان پشته<sup>۱۰۰</sup> کردن آن برای عمیق ساختن ویژگی‌ها بین مقیاس وجود دارد؛ جزئیات لایه و معماری استفاده کننده از آن در بخش ۲-۷ مورد بررسی قرار گرفته و در بخش ۳-۷ نیز نتایج حاصل از استفاده‌ی لایه‌ی هرم ویژگی دوطرفه<sup>۱۰۱</sup> گزارش و با شبکه‌های دیگر تشخیص اشیاء مقایسه خواهد شد.

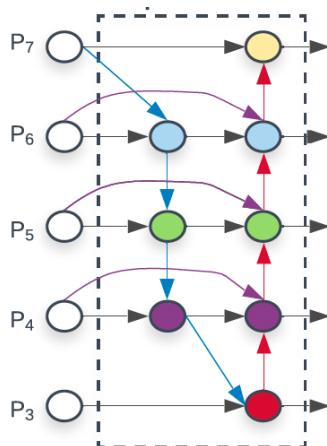
<sup>۹۹</sup> Mingxing Tan

<sup>۱۰۰</sup> Stack

<sup>۱۰۱</sup> Bi-feature pyramid network(BiFPN)

## ۷ - ۲ - هرم ویژگی دو طرفه

هرم ویژگی دو طرفه ارائه شده توسط تان و همکاران [۱۷] در دو گام ویژگی مقیاس‌های مختلف را با یکدیگر ادغام کرده و اطلاعات هر یک در در همهٔ مقیاس‌ها تاثیر می‌دهد. طرحواره هرم ویژگی دو طرفه در شکل ۲-۷ قابل ملاحظه می‌باشد.



شکل ۲-۷: طرحواره هرم ویژگی دو طرفه [۱۷]

انتقال ویژگی بین مقیاس‌های مختلف در دو گام صورت می‌گیرد؛ در گام اول ابتدا بصورت بالا به پایین ویژگی‌های صفحه کوچکتر از طریق فرآنمونه برداری<sup>۱۰۲</sup> هم ابعاد با مقیاس بعدی شده و سپس توسط عمل کانولوشن با آن ترکیب شده و ویژگی‌های میانی را به وجود می‌آورد. البته لازم به ذکر است که ترکیب‌های بین ویژگی‌ها همگی بصورت وزن‌دار انجام پذیرفته و وزن‌ها نیز می‌توانند قابل یادگیری باشند.

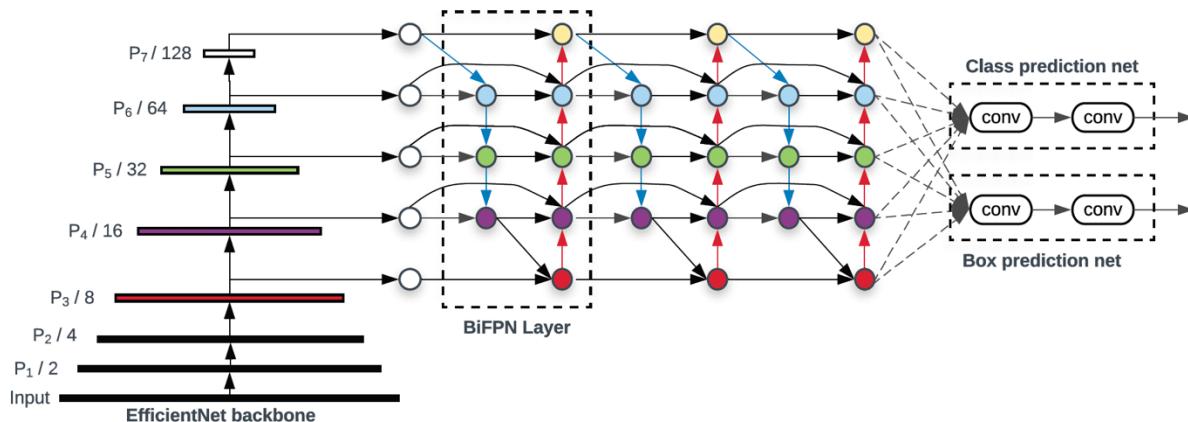
در گام دوم بصورت پایین به بالا ویژگی‌های صفحه بزرگتر از طریق فرآنمونه برداری<sup>۱۰۳</sup> هم ابعاد با مقیاس قبلی خود شده و سپس توسط عمل کانولوشن با ویژگی میانی و ورودی متناظر ترکیب شده و ویژگی‌های خروجی را به وجود می‌آورد. این فرآیند و لایه را می‌توان بصورت پشت سرهم قرار داده و صفحات ویژگی مفیدتر و عمیق‌تری را نیز حاصل نمود.

<sup>102</sup> Up-sampling

<sup>103</sup> Down-sampling

اتصال ورودی یک مقیاس علاوه بر ویژگی میانی<sup>۱۰۴</sup> همان مقیاس باعث می‌شود که این اختیار به شبکه و آموزش داده شود که تصمیم بگیرد چه مقیاسی و از چه ورودی‌هایی با چه وزن‌هایی تاثیر بپذیرد که همین امر بهبود استخراج ویژگی و در نتیجه تشخیص اشیاء را بدنبال خواهد داشت.

حال بایستی از لایه‌ی هرم ویژگی دو طرفه [۱۷] برای استخراج ویژگی در معماری یک شبکه تشخیص اشیا بهره گرفت تا بتوان تاثیر آن و قدرت ویژگی‌های استخراجی ارزیابی نمود. تان و همکاران [۱۷] معماری شبکه تشخیص کارآمد اشیا را بصورت شکل ۷-۳ ارائه کرده اند که در آن از هرم ویژگی دو طرفه استفاده کرده اند.



شکل ۷-۳: معماری شبکه تشخیص کارآمد اشیاء [۱۷]

در معماری شبکه تشخیص کارآمد اشیاء [۱۲] از ResNet-50 [۱۳] به عنوان ستون استخراج ویژگی استفاده شده و خروجی لایه‌های آن در مقیاس‌های مختلف به عنوان ورودی اولین هرم ویژگی دو طرفه در نظر گرفته شده است. سپس دو لایه‌ی هرم ویژگی دو طرفه‌ی دیگر نیز در ادامه پشت سر هم پشته شده و در نهایت توسط لایه‌های پیچشی مانند شبکه رتینا [۱۲] اقدام به تشخیص جعبه‌های مرزی می‌شود.

### ۷ - ۳ - گزارش نتایج

در این بخش به نتایج و اثرات حاصل از استفاده‌ی هرم ویژگی دو طرفه و شبکه تشخیص کارآمد اشیاء [۱۷] پرداخته می‌شود که در جدول ۷-۱ آورده شده است. قابل ملاحظه است شبکه مذکور توانسته در بهترین نسخه خود از تمامی نتایج پیشین عملکرد مطلوب‌تری را ارائه کرده و به دقت بالای ۵۰ درصد در مجموعه داده‌ی MS COCO [۱۱] دست یابد.

<sup>104</sup> Middle

بهینه‌ساز مورد استفاده در آموزش تان و همکاران [۱۷] نزول گرادیان تصادفی بوده و توسط ۱۱۸ هزار تصویر آموزش یادگیری انجام پذیرفته است.

**جدول ۷-۱: مقایسه‌ی عملکرد و جزئیات اجرایی شبکه تشخیص کارآمد اشیاء با سایر شبکه‌ها** [۱۷]

Model	tet-dev			val AP	Params	Ratio	FLOPs	Ratio	Latency	
	AP	AP <sub>50</sub>	AP <sub>75</sub>						GPU <sub>ms</sub>	CPU <sub>s</sub>
<b>EfficientDet-D0 (512)</b>	<b>33.8</b>	<b>52.2</b>	<b>35.8</b>	<b>33.5</b>	<b>3.9M</b>	<b>1x</b>	<b>2.5B</b>	<b>1x</b>	<b>16</b>	<b>0.32</b>
YOLOv3 [31]	33.0	57.9	34.4	-	-	-	71B	28x	51 <sup>†</sup>	-
<b>EfficientDet-D1 (640)</b>	<b>39.6</b>	<b>58.6</b>	<b>42.3</b>	<b>39.1</b>	<b>6.6M</b>	<b>1x</b>	<b>6.1B</b>	<b>1x</b>	<b>20</b>	<b>0.74</b>
RetinaNet-R50 (640) [21]	37.0	-	-	-	34M	6.7x	97B	16x	27	2.8
RetinaNet-R101 (640)[21]	37.9	-	-	-	53M	8.0x	127B	21x	34	3.6
<b>EfficientDet-D2 (768)</b>	<b>43.0</b>	<b>62.3</b>	<b>46.2</b>	<b>42.5</b>	<b>8.1M</b>	<b>1x</b>	<b>11B</b>	<b>1x</b>	<b>24</b>	<b>1.2</b>
RetinaNet-R50 (1024) [21]	40.1	-	-	-	34M	4.3x	248B	23x	51	7.5
RetinaNet-R101 (1024) [21]	41.1	-	-	-	53M	6.6x	326B	30x	65	9.7
ResNet-50 + NAS-FPN (640) [8]	39.9	-	-	-	60M	7.5x	141B	13x	41	4.1
<b>EfficientDet-D3 (896)</b>	<b>45.8</b>	<b>65.0</b>	<b>49.3</b>	<b>45.9</b>	<b>12M</b>	<b>1x</b>	<b>25B</b>	<b>1x</b>	<b>42</b>	<b>2.5</b>
ResNet-50 + NAS-FPN (1024) [8]	44.2	-	-	-	60M	5.1x	360B	15x	79	11
ResNet-50 + NAS-FPN (1280) [8]	44.8	-	-	-	60M	5.1x	563B	23x	119	17
ResNet-50 + NAS-FPN (1280@384)[8]	45.4	-	-	-	104M	8.7x	1043B	42x	173	27
<b>EfficientDet-D4 (1024)</b>	<b>49.4</b>	<b>69.0</b>	<b>53.4</b>	<b>49.0</b>	<b>21M</b>	<b>1x</b>	<b>55B</b>	<b>1x</b>	<b>74</b>	<b>4.8</b>
AmoebaNet+ NAS-FPN +AA(1280)[42]	-	-	-	48.6	185M	8.8x	1317B	24x	259	38
<b>EfficientDet-D5 (1280)</b>	<b>50.7</b>	<b>70.2</b>	<b>54.7</b>	<b>50.5</b>	<b>34M</b>	<b>1x</b>	<b>135B</b>	<b>1x</b>	<b>141</b>	<b>11</b>
<b>EfficientDet-D6 (1280)</b>	<b>51.7</b>	<b>71.2</b>	<b>56.0</b>	<b>51.3</b>	<b>52M</b>	<b>1x</b>	<b>226B</b>	<b>1x</b>	<b>190</b>	<b>16</b>
AmoebaNet+ NAS-FPN +AA(1536)[42]	-	-	-	50.7	209M	4.0x	3045B	13x	608	83
<b>EfficientDet-D7 (1536)</b>	<b>52.2</b>	<b>71.4</b>	<b>56.3</b>	<b>51.8</b>	<b>52M</b>	<b>1x</b>	<b>325B</b>	<b>1x</b>	<b>262</b>	<b>24</b>

نکته حائز اهمیت در عملکرد شبکه تشخیص کارآمد اشیاء [۱۷] مدت زمان اجرایی پایین آن توأم با تعداد پارامتر آموزشی کمتر نسبت به سایر شبکه‌های تشخیص اشیای تک مرحله‌ای می‌باشد؛ این امر نشان می‌دهد استفاده از لایه‌ی هرم ویژگی دو طرفه می‌تواند علاوه بر بهبود دقیقت تشخیص اشیاء، در زمان اجرایی و حجم شبکه نیز اثر مطلوب داشته باشد.

## **فصل هشتم جمع‌بندی و مراجع**

## ۱ - بحث

تشخیص اشیاء در تصاویر یکی از مباحث داغ و چالشی حوزه بینایی کامپیوتر است که در کاربردهای گوناگون مورد استفاده قرار می‌گیرد. در فصل اول گامها و رویکردهای تشخیص اشیاء مورد مطالعه قرار گرفت و بیان شد که مقصود از تک مرحله‌ای بودن تشخیص اشیاء در تصاویر عبارت است از اینکه تصویر ورودی صرفاً یکبار مورد پردازش و بررسی قرار گرفته و نتیجه تشخیص حاصل شود. انگیزه‌های متعددی برای مطالعه‌ی مدل‌های تشخیص اشیایی تک مرحله‌ای وجود دارد که اهم آن زمان اجرایی کوتاه و ساختار نزدیک آن به سیستم بینایی انسان است که می‌تواند صرفاً با یک نگاه اشیایی صحنه را تشخیص دهد [۴].

شبکه‌ی یولو [۴] به عنوان اولین شبکه‌ی عمیق اقدام به تشخیص اشیای موجود در تصویر کرد که توانست با ارائه‌ی جعبه‌های مرزی بالقوه چندین برابر سریع‌تر از شبکه‌های تشخیص دو مرحله‌ای عمیق عمل کرده و زمینه را برای پژوهش در این عرصه فراهم آورد.

در شبکه‌ی مذکور فوق دقت تشخیص با وجود اندکی ضعف نسبت به مدل‌های دو مرحله‌ای، رقابت نزدیکی با آنان داشت. در شبکه‌ی یولو [۴] فرآیند پیش‌بینی توسط شبکه‌های تمام متصل انجام می‌پذیرفت که شبکه‌ی اس‌اس‌دی [۵] با تغییر آن به شبکه‌های پیچشی توانست به دقت بالاتری در تشخیص اشیاء رسید که تمرکز آن بر تشخیص چندین شیء بصورت توامان در تصویر ورودی بود.

چالشی که در آموزش شبکه‌های عمیق برای تشخیص اشیاء وجود داشته و دقت پیش‌بینی را محدود می‌ساخت، نامتعادل بودن تعداد نواحی پس‌زمینه و اشیاء در حین آموزش شبکه بود؛ شبکه رتینا [۱۲] با معرفیتابع هزینه کانونی (مرکزی) که منطبق بر تابع آنتروپی متقابل بود، توانست این مشکل را با دادن وزن به نمونه‌ها مرتفع سازد تا جایی که شبکه‌های عمیق متعددی از آن استفاده کرده‌اند.

ایده‌ی بعدی برای بهبود عملکرد تشخیص اشیاء در شبکه‌های عمیق عدم استفاده از جعبه‌های مرزی پیشفرض و بالقوه بود که توسط شبکه‌های گوشه محور [۱۵] و مرکز محور [۱۴] بیان شد. شبکه‌های مذکور مستقیماً و توسط مدل‌های رگرسیونی تلاش کردند که مرکز و گوش‌های جعبه‌ی مرزی را مستقیماً تخمین زده و با ترکیب نتایج اقدام به تشخیص و شناسایی اشیاء کنند که موفق بوده و توانستند ضمن کاهش پیچیدگی، دقت تشخیص و شناسایی را بهبود دهند.

در تلاش دیگری برای عدم استفاده از جعبه‌های مرزی، شبکه‌ی تشخیص شیء کاملاً پیچشی [۱۶] اقدام به تخمین جعبه‌های مرزی بصورت مستقیم و به ازای هر پیکسل کرد که در ادامه با ترکیب نتایج پیکسل‌ها توانست میانگین دقت تشخیص و شناسایی را مجدد افزایش دهد تا جایی که در رقابت با مدل‌های تشخیص دو مرحله‌ای پیشی گیرد.

در مطالعه دیگری که توسط تان و همکاران [۱۷] انجام و شبکه‌ی تشخیص کارآمد اشیاء پیشنهاد شد که در آن هدف بهبود ویژگی‌های استخراجی در هرم ویژگی بود که توانست علاوه بر کاهش بار محاسباتی، پیچیدگی و تعداد پارامترهای آموزشی، میانگین دقت را بالا برد. اساس راهکار پیشنهادی مبتنی بر دو طرفه ساختن اتصال ویژگی‌ها در هرم ویژگی با ایجاد ویژگی‌ها میانی بود.

قابل ملاحظه است که از اولین تلاش برای تشخیص اشیاء بصورت تک مرحله‌ای و مبتنی بر شبکه‌های عمیق همواره چالش‌ها و مسائل متعددی مطرح بوده و پژوهشگران در گذر زمان در تلاش برای ارائه راهکار برای هر یک بوده اند. این چالش‌ها بحث ساختار شبکه‌ها، استفاده از لایه‌ها و مازویل‌های گوناگون، تمرکز و اصلاح توابع هزینه، تغییر رویکرد در تخمین جعبه‌های مرزی و... را در بر می‌گیرد و مبرهن است که ارائه راه حل برای هر یک می‌تواند باعث بهبود در مدل‌های تک مرحله‌ای عمیق می‌شود. مطالعه‌ی هر یک از چالش‌ها و راهکارهای پیشنهادی و تلاش در ارتقای آن می‌تواند یک زمینه مطالعاتی جدید را فراهم آورد.

در این گزارش مطالعاتی تلاش شد سیر شبکه‌های تشخیص اشیای تک مرحله‌ای مبتنی بر شبکه‌های عمیق مورد بررسی قرار گیرد و نحوه بیان آن بصورت خط زمانی <sup>۱۰۵</sup> انجام پذیرد تا خواننده بتواند روند تحولات شامل تغییرات دقت پیش‌بینی و بار محاسباتی و پیچیدگی را پیگیری و دنبال نماید.

<sup>105</sup> Time-line

## ۸ - مراجع

- [1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [2] 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.
- [3] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [4] 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.
- [5] W. Liu *et al.*, “SSD: Single Shot Multibox Detector,” in *European conference on computer vision*, 2016, pp. 21–37.
- [6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *ArXiv Prepr. ArXiv14091556*, 2014.
- [7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.” [Online]. Available: <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>
- [8] H. Cai, Q. Wu, T. Corradi, and P. Hall, “The cross-depiction problem: Computer vision algorithms for recognising objects in artwork and in photographs,” *ArXiv Prepr. ArXiv150500110*, 2015.
- [9] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [10] 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.
- [11] T.-Y. Lin *et al.*, “Microsoft coco: Common objects in context,” in *European conference on computer vision*, 2014, pp. 740–755.
- [12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

- [14] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, “CenterNet: Keypoint triplets for object detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6569–6578.
- [15] H. Law and J. Deng, “CornerNet: Detecting objects as paired keypoints,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.
- [16] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9627–9636.
- [17] 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.