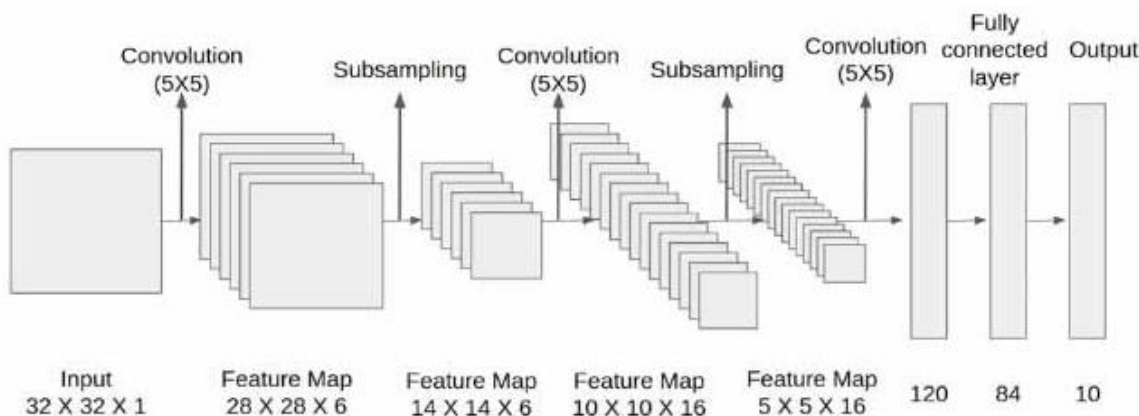
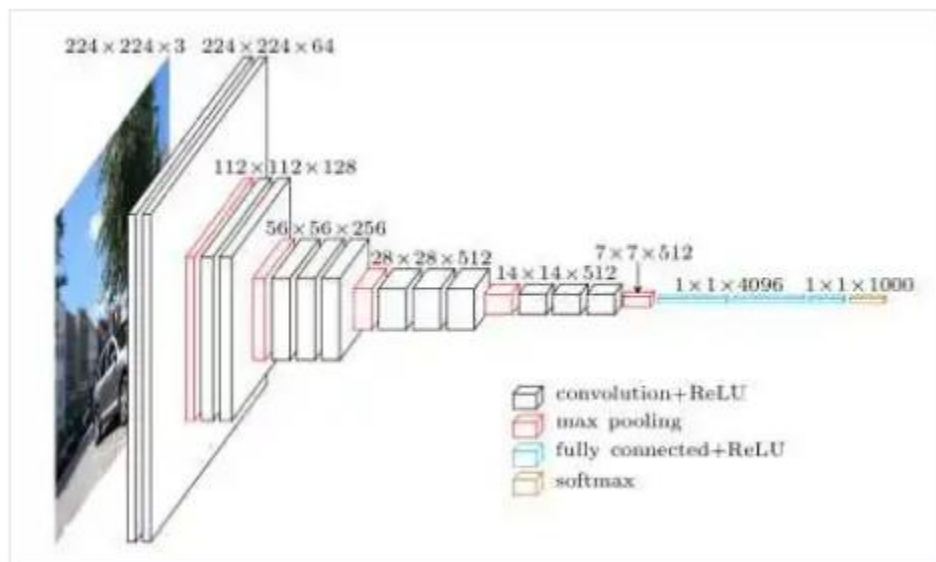


شبکه LeNet به بخش های مختلفی تقسیم می شود. لایه ها به صورت جداگانه به صورت زیر می باشند:

- لایه input یا ورودی: در این لایه تصاویر بعد از پیش پردازش به شبکه LeNet وارد می شود. بر اساس اولین مدل ارائه شده در مقاله Yann LeCun این مرحله سایز تصاویر 32×32 است.
- لایه کانولوشن: در این مرحله بر روی تصاویر ورودی عملیات کانولوشن انجام می شود. سایز Kernel در این عملیات 5×5 بوده و از 6 کرنل مختلف استفاده خواهد شد. در پایان مرحله تصاویر ما سایزی برابر با 28×28 داشته که با توجه به Kernel اعمال شده، سایز پایانی خروجی این مرحله $28 \times 28 \times 6$ می باشد.
- عملیات Pooling: مرحله بعد عملیات Pooling می باشد. این کار برای کاهش سایز شبکه استفاده می شود. پس از پایان این مرحله سایز تصاویر نصف می شود. سایز خروجی $14 \times 14 \times 6$ می باشد.
- لایه کانولوشن: در این مرحله مجدداً عملیات کانولوشن انجام می شود. البته این بار بر روی خروجی مرحله قبل انجام می شود. سایز Kernel در این مرحله 5×5 بوده و این بار 16 فیلتر متفاوت استفاده خواهد شد. سایز خروجی برابر با $10 \times 10 \times 16$ می باشد.
- عملیات Pooling: مجدداً عملیات Pooling انجام می شود. پس از پایان این مرحله سایز تصاویر نصف می شود. سایز خروجی $5 \times 5 \times 16$ می باشد.
- لایه کانولوشن: این مرحله را مرحله آخر می گوئیم چراکه مرحله بعد اتصال خروجی به شبکه عصبی است. باز هم یک لایه کانولوشن داریم با kernel به سایز 5×5 و تعداد 120 فیلتر. خروجی این مرحله $1 \times 1 \times 120$ می باشد.
- لایه تماماً متصل یا Fully Connected Layer: در این بخش خروجی مرحله قبل را به عنوان ورودی شبکه عصبی قرار می دهیم. شبکه عصبی این مرحله تماماً متصل بوده و شامل 84 نرون می باشد.
- لایه خروجی: تعداد اعدادی که شبکه ما شناسایی می کند 10 عدد است یعنی اعداد 0، 1، 2، 3، 4، 5، 6، 7، 8، 9. در نهایت تعداد نرون لایه خروجی برای شبکه عصبی 10 عدد است.



VGG NET

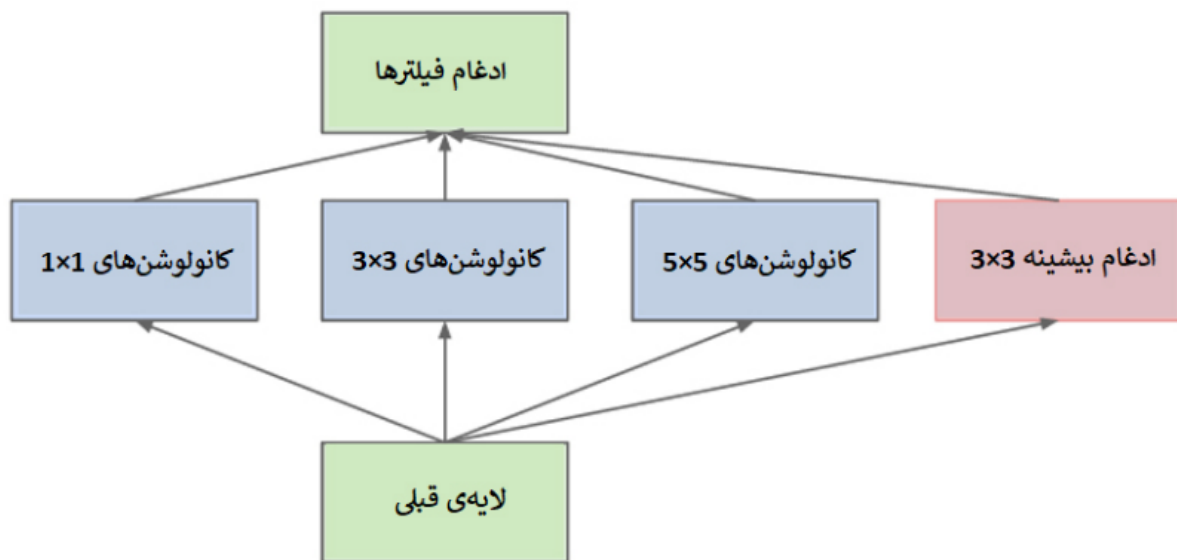


همانطور که در تصویر مشخص است، VGG شامل یک سری از لایه‌های محاسباتی (Convolutional) است که پشت آن‌ها لایه‌های جمع‌کننده (pooling) وجود دارند که لایه‌ها را کوچکتر می‌کنند. این گروه در تحقیقات خود شبکه‌های مختلفی را مطرح کرده‌است که هر کدام آن‌ها عمق این معماری را تغییر می‌دهند.

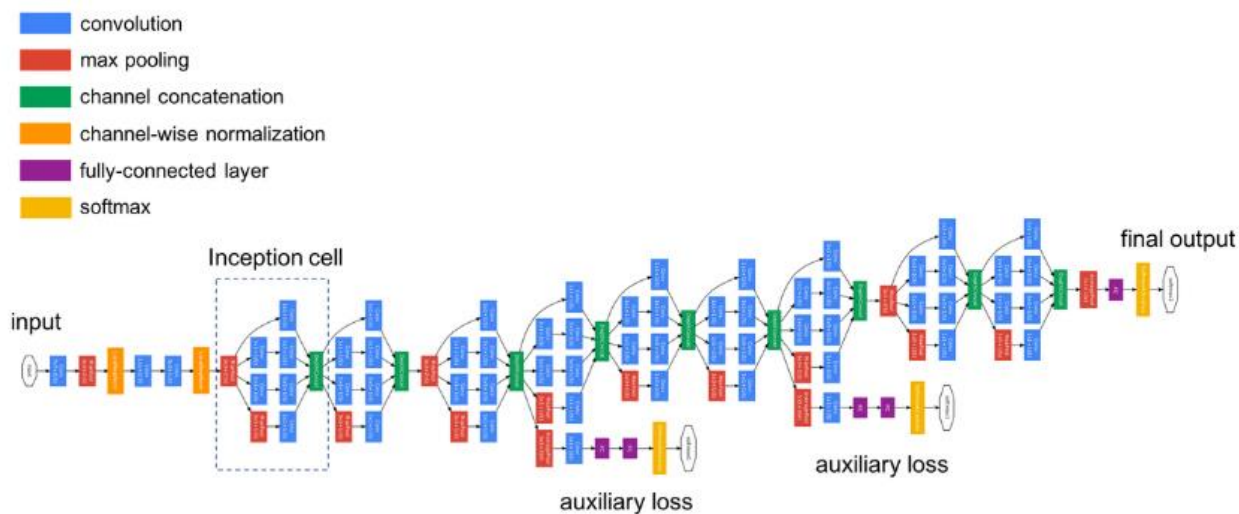
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Google net

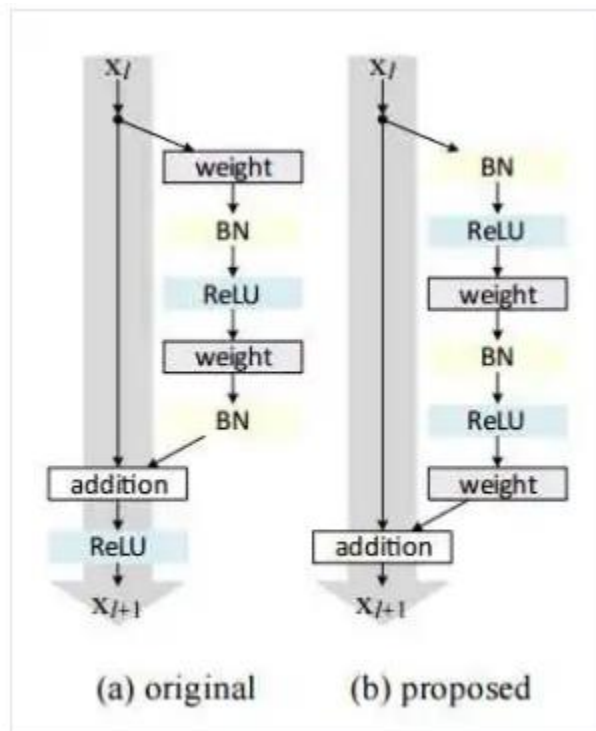
GoogleNet یک معماری است که توسط محققین گوگل طراحی شده است. GoogleNet با ارائه قویترین مدل، برنده «ImageNet 2014» بود. در این معماری، علاوه بر عمق بیشتر (دارای 22 لایه است، در مقایسه با VGG که دارای 19 لایه است)، محققان یک رویکرد جدید نیز به نام ماژول آغازین (Inception Module) ارائه دادند.



همانطور که در تصویر بالا می بینید، این ماژول یک تغییر بزرگ نسبت به معماری های ترتیبی است که در بالاتر دیدیم. در یک لایه، چندین نوع «feature extractor» لایه هایی که مقادیر ورودی را دریافت کرده، و به نوعی داده برای محاسبات تبدیل می کنند وجود دارند. در شبکه ای که خود در حال یادگیری است و باید از گزینه های مختلفی برای حل وظایف استفاده کند، این نوع لایه بندی، به طور غیر مستقیم به عملکرد بهتر شبکه کمک می کند. این ماژول می تواند از ورودی ها مستقیماً در محاسبات خود استفاده کند، یا آن ها را مستقیماً جمع بندی کند.

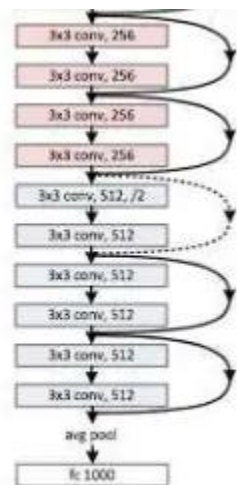
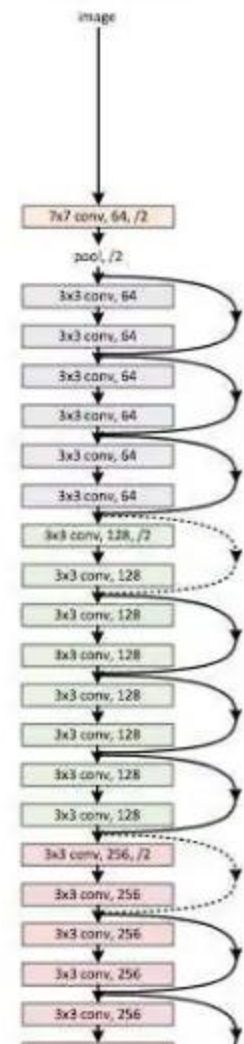


ResNet یکی از معماری‌های بزرگ است که نشان می‌دهد یک معماری یادگیری عمیق تا چه حد می‌تواند عمیق باشد (ResNet). که مخفف Residual Networks است) شامل چندین ماژول رسوبی است که بر روی هم سوار شده‌اند، که در واقع ساختمان اصلی معماری ResNet را تشکیل می‌دهند. تصویر زیر یک نمای کلی از ماژول رسوبی را نشان می‌دهد.



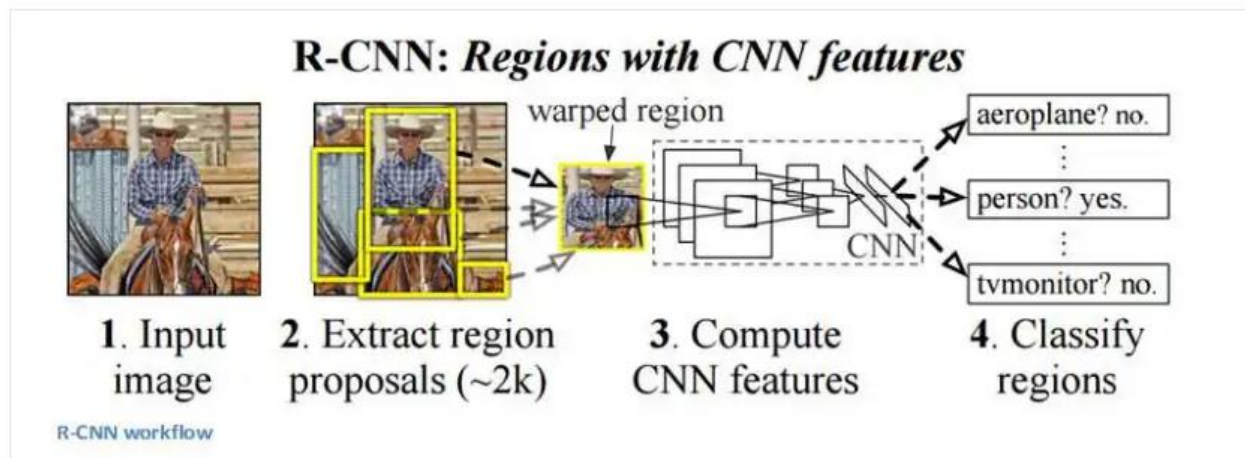
به بیان ساده، ماژول رسوبی دو راه دارد، یا می‌تواند یک سری عملیات را بر روی ورودی انجام دهد، یا تمام این مراحل را رد کند. همانند GoogleNet، این ماژول‌های رسوبی بر روی یک دیگر سوار شده‌اند تا یک شبکه کامل را تشکیل دهند.

34-layer residual

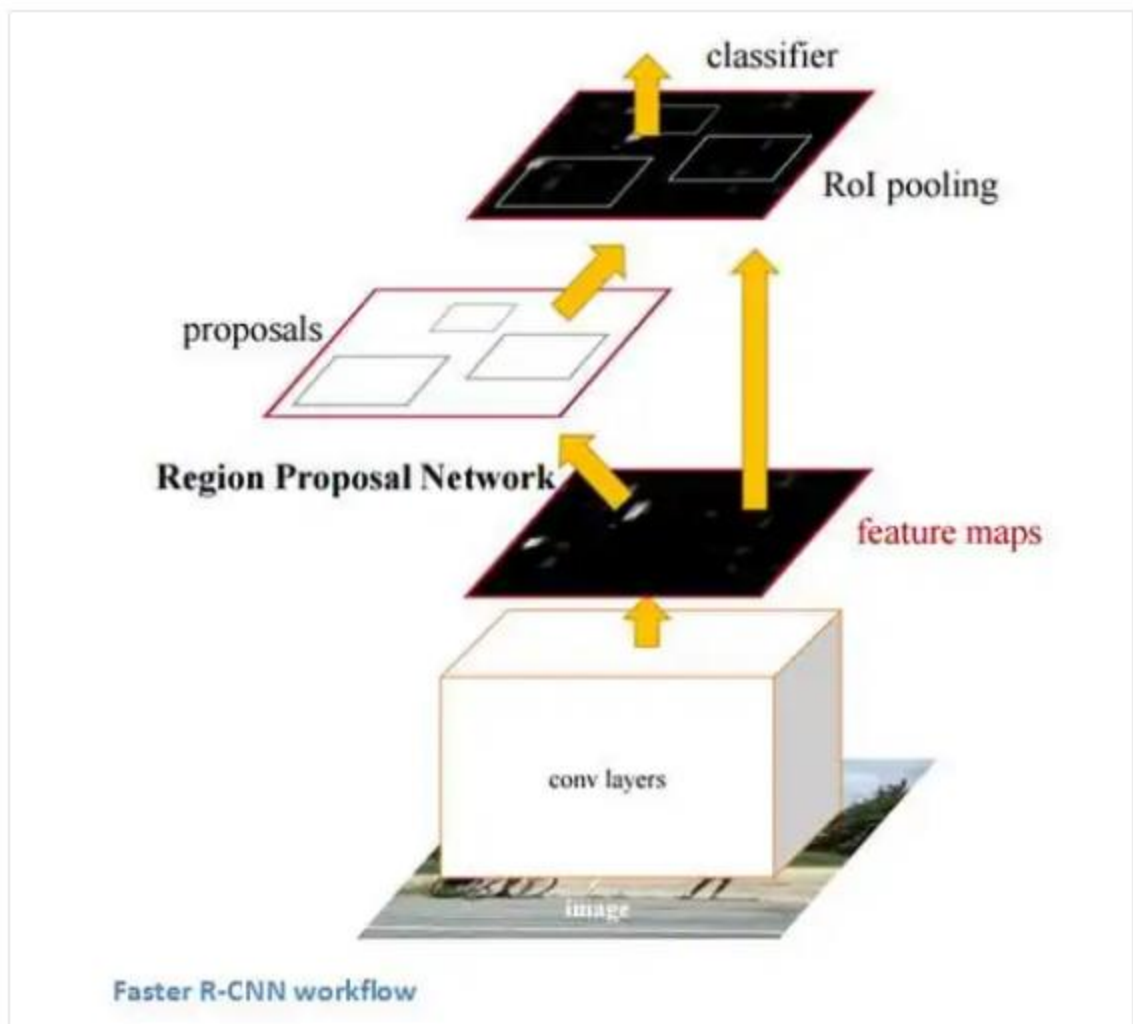


RCNN

معماری (Region Based CNN بر پایه محل قرارگیری) تأثیرگذارترین معماری یادگیری عمیق است که تا به حال برای حل مسائل تشخیص اشیاء استفاده شده است. نحوه کار این معماری اینگونه است که سعی می‌کند دور تمام اشیای حاضر در تصویر یک جعبه بکشد و سپس شیء داخل تصویر را شناسایی کند. نحوه کار آن در تصویر زیر آمده است:

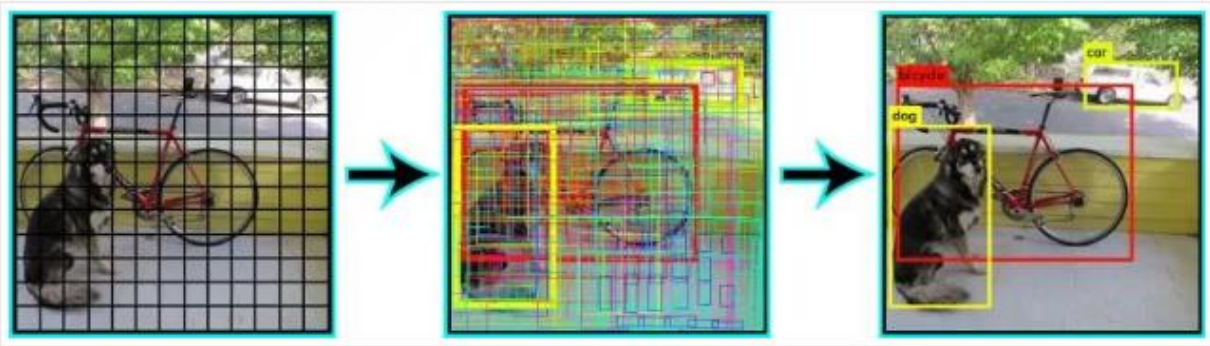


ساختار این معماری نیز در تصویر زیر آمده است:

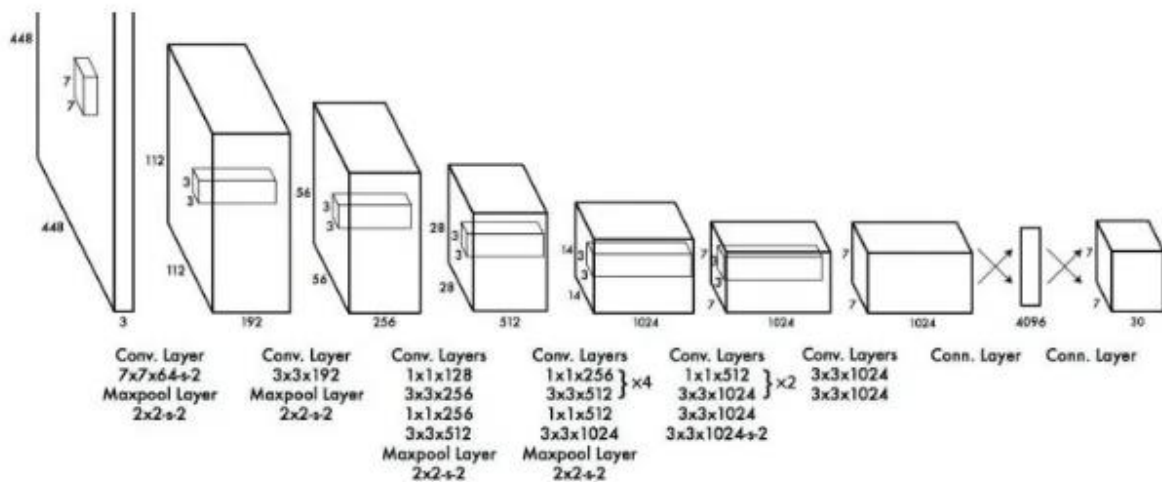


YOLO

YOLO همان‌طور که در تصویر زیر مشاهده می‌کنید، این الگوریتم ابتدا تصویر را به بخش‌های مختلف تقسیم می‌کند و هر بخش را علامت‌گذاری می‌کند، سپس الگوریتم شناسایی را به صورت موازی برای تمامی این بخش‌ها اجرا می‌کند تا ببیند هر بخش به کدام دسته‌بندی تعلق می‌گیرد. بعد از شناسایی کامل اشیاء، آن‌ها را به هم متصل می‌کند تا دو هر شیء اصلی یک جعبه باشد.



در زیر تصویری از معماری YOLO را مشاهده می‌کنید.



Mask RCNN

معماری Mask R-CNN یک مدل تعمیم یافته از Faster R-CNN است که ما در پست قبلی به آن اشاره کردیم. جهت یادآوری معماری Faster R-CNN به شرح زیر بود: لایه های کانولوشنی: تصویر ورودی از چندین لایه کانولوشن عبور داده می شود تا یک نقشه ویژگی استخراج شود.

