Object Detection using YOLOv3

 [Nima Dorostkar](https://github.com/nimadorostkar) AUGUST 20, 2019

YOLOv3 is the latest variant of a popular object detection algorithm YOLO – You Only Look Once. The published model recognizes 80 different objects in images and videos, but most importantly it is super fast and nearly as accurate as Single Shot MultiBox (SSD).

Starting with OpenCV, you can easily use YOLOv3 models in your own OpenCV application.

How does YOLO work ?

We can think of an object detector as a combination of a object locator and an object recognizer.

In traditional computer vision approaches, a sliding window was used to look for objects at different locations and scales. Because this was such an expensive operation, the aspect ratio of the object was usually assumed to be fixed.

Early Deep Learning based object detection algorithms like the R-CNN and Fast R-CNN used a method called [Selective Search](https://learnopencv.com/selective-search-for-object-detection-cpp-python/) to narrow down the number of bounding boxes that the algorithm had to test.

Another approach called Overfeat involved scanning the image at multiple scales using sliding windows-like mechanisms done convolutionally.

This was followed by Faster R-CNN that used a Region Proposal Network (RPN) for identifying bounding boxes that needed to be tested. By clever design the features extracted for recognizing objects, were also used by the RPN for proposing potential bounding boxes thus saving a lot of computation.

YOLO on the other hand approaches the object detection problem in a completely different way. It forwards the whole image only once through the network. SSD is another object detection algorithm that forwards the image once though a deep learning network, but YOLOv3 is much faster than SSD while achieving very comparable accuracy. YOLOv3 gives faster than realtime results on a M40, TitanX or 1080 Ti GPUs.

Lets see how YOLO detects the objects in a given image.

First, it divides the image into a 13×13 grid of cells. The size of these 169 cells vary depending on the size of the input. For a 416×416 input size that we used in our experiments, the cell size was 32×32. Each cell is then responsible for predicting a number of boxes in the image.

For each bounding box, the network also predicts the confidence that the bounding box actually encloses an object, and the probability of the enclosed object being a particular class.

Most of these bounding boxes are eliminated because their confidence is low or because they are enclosing the same object as another bounding box with very high confidence score. This technique is called non-maximum suppression.

The authors of [YOLOv3](https://pjreddie.com/media/files/papers/YOLOv3.pdf), Joseph Redmon and Ali Farhadi, have made YOLOv3 faster and more accurate than their previous work [YOLOv2](https://pjreddie.com/media/files/papers/YOLO9000.pdf). YOLOv3 handles multiple scales better. They have also improved the network by making it bigger and taking it towards residual networks by adding shortcut connections.

تشخیص اشیا با استفاده از YOLOv3

YOLOv3 آخرین ورژن از الگوریتم محبوب تشخیص اشیا YOLO است (شما فقط یک بار نگاه می کنید). مدل منتشر شده ۸۰ شی مختلف را در تصاویر و فیلم ها تشخیص می دهد، اما مهمتر از همه سرعت بسیار زیاد آن که تقریباً مثل تک شات MultiBox (SSD) است.

شروع با OpenCV ، شما به راحتی می توانید از مدل های YOLOv3 در برنامه OpenCV خود استفاده کنید.

YOLO چگونه کار می کند؟

ما می توانیم یک تشخیص گر اشیا را ترکیبی از مکان یاب اشیا و شناسایی کننده اشیا بدانیم. در روش های سنتی بینایی کامپیوتر، از یک پنجره کشویی برای جستجوی اشیا در مکان ها و مقیاس های مختلف استفاده می شد. از آنجا که این عملیات گران قیمت بود ، معمولاً فرض بر این بود که نسبت ابعاد جسم ثابت است.

پیش از این الگوریتم های تشخیص اشیا مبتنی بر یادگیری عمیق مانند R-CNN و Fast R-CNN از روشی به نام جستجوی انتخابی استفاده می کردند تا تعداد کادرهای محصور کننده را که الگوریتم باید آزمایش کند ، کاهش دهند. رویکرد دیگری به نام Overfeat شامل اسکن تصویر در مقیاس های مختلف با استفاده از مکانیزم های کشویی مانند پنجره ها که به صورت مکانیزم های پیچشی انجام می شود.

به دنبال آن Faster R-CNN که از شبکه Proposal Network (RPN) برای شناسایی کادرهای محصور کننده که باید رویشان آزمایش انجلم شود ، استفاده شد. با طراحی هوشمندانه ویژگی های استخراج شده برای شناسایی اشیا ، همچنین توسط RPN برای پیشنهاد کادرهای محصور کننده بالقوه مورد استفاده قرار گرفت و در نتیجه در محاسبات زیادی صرفه جویی شد.

از طرف دیگر YOLO با استفاده از روشی کاملاً متفاوت مشکل تشخیص اشیا را حل میکند. فقط یکبار کل تصویر را از طریق شبکه ارسال می کند. SSD یکی دیگر از الگوریتم های تشخیص اشیا است که با وجود داشتن یک شبکه یادگیری عمیق ، تصویر را یک بار به جلو هدایت می کند ، اما YOLOv3 بسیار سریعتر از SSD است و در عین حال به دقت قابل مقایسه ای دست می یابد. YOLOv3 نتیجه ای سریعتر از نتایج لحظه ای در پردازنده های گرافیکی M40 ، TitanX یا 1080 Ti ارائه می دهد.

بیایید ببینیم YOLO چگونه اشیا را در یک تصویر مشخص تشخیص می دهد.

ابتدا تصویر را به شبکه 13 × 13 سلول تقسیم می کند. اندازه این 169 سلول بسته به اندازه ورودی متفاوت است. برای اندازه ورودی ۴۱۶ در ۴۱۶ که در آزمایشات خود استفاده کردیم ، اندازه سلول ها 32 × 32 بود. سپس هر سلول مسئول پیش بینی تعدادی از کادرهای محصور کننده است.

برای هر کادر ، شبکه همچنین اطمینان حاصل میکند که کادرهای محصور کننده در واقع یک شی را محصور کرده و احتمال اینکه شی محصور شده یک کلاس خاص باشد.

بیشتر این کادر ها از بین می روند زیرا صلاحیت آنها پایین است یا به همین دلیل همان شی کادر اتصال دیگری را با نمره اطمینان بسیار بالا محصور می کنند. این روش سرکوب غیر حداکثر نامیده می شود.

نویسندگان YOLOv3 ، جوزف ردمون و علی فرهادی ، YOLOv3 را سریعتر و دقیق تر از کار قبلی خود YOLOv2 ساخته اند. YOLOv3 مقیاس های مختلف را بهتر کنترل می کند. آنها همچنین با بزرگتر کردن شبکه و با اضافه کردن اتصالات میانبر ، آن را به سمت شبکه های باقیمانده سوق داده اند.