YOLOv3: An Incremental Improvement

備考

元論文

著者

Joseph Redmon, Ali Farhadi

掲載

"Yolov3: An incremental improvement,".arXiv preprint arXiv:1804.02767, 2018.

Abstract

YOLOのアップデートをご紹介します！私たちはそれをより良くするために小さな設計変更をたくさん加えました。また、かなり膨張したこの新しいネットワークをトレーニングしました。前回より少し大きいですが、より正確です。まだ高速ですから心配しないでください。 320で320 YOLOv3は22msで28.2 mAPで動作し、SSDと同じくらい正確ですが3倍高速です。古い.5 IOU mAP検出メトリックを見ると、YOLOv3は非常に優れています。 Titan Xでは、51 msで57.9 AP50を達成しています。RetinaNetで198 msで57.5 AP50と比較しています。同様のパフォーマンスですが、3.8倍高速です。いつものように、すべてのコードはhttps://pjreddie.com/yolo/でオンラインです。

1. Introduction

たまに1年間電話をかけたりするよね？今年はあまり研究をしませんでした。Twitterで多くの時間を費やした。昨年から人気だったGANを少しいじってみました。 [12] [1]。なんとかYOLOを改善しました。しかし、正直なところ、非常に興味深いものは何もありません。それを改善する小さな変更の集まりです。他の人の調査にも少し手伝いました。

私たちをここにもたらしています。カメラ対応の期限[4]があり、YOLOに対して行ったランダムな更新の一部を引用する必要がありますが、ソースがありません。だからTECH REPORTの準備をしましょう！

技術レポートの優れた点は、紹介が不要なことです。なぜ私たちがここにいるのかは誰もが知っています。したがって、この紹介の終わりは、残りの論文の道しるべとなります。まず、YOLOv3との取引について説明します。次に、その方法を説明します。うまくいかなかったいくつかのことについてもお知らせします。最後に、これが何を意味するかを考えます。

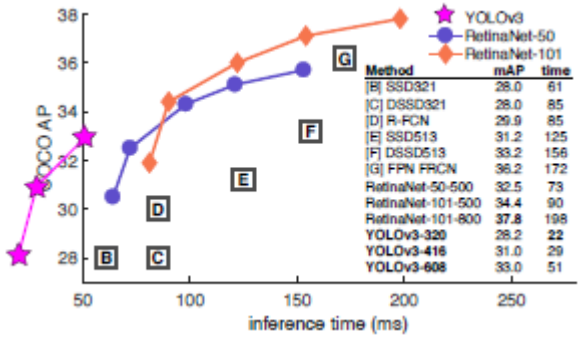

図1. Focal Loss論文[9]からこの図を採用しました。 YOLOv3は、同等のパフォーマンスを持つ他の検出方法よりも大幅に高速に実行されます。 M40またはTitan Xからの時間、それらは基本的に同じGPUです。

2. The Deal

これがYOLOv3との取引です。主に他の人から良いアイデアを取り入れました。また、他のネットワークよりも優れた新しい分類ネットワークをトレーニングしました。すべてを理解できるように、システム全体を最初から説明します。

2.1. Bounding Box Prediction

YOLO9000に従って、私たちのシステムは、次元クラスター(集団)をアンカーボックスとして使用して境界ボックスを予測します[15]。ネットワークは、境界ボックスごとに4つの座標、t_x,t_y,t_w,t_hを予測します。セルが画像の左上隅から(c_x, c_y)だけオフセットされ、前の境界ボックスの幅と高さが$p_w、p_h$の場合、予測は次のように対応します。

$$ b_x = \sigma (t_X) + c_x \ b_y = \sigma (t_y) + c_y \ b_w = p_w e^{t_w} \ b_h = p_h e^{t_h} $$

トレーニング中、誤差の二乗和を使用します。ある座標予測のグラウンドトゥルースが$\hat{t}*$の場合、グラジエントはグラウンドトゥルース値（グラウンドトゥルースボックスから計算）から予測$*hat{t}* -t_*$を差し引いたものになります。このグラウンドトゥルース値は、上記の方程式を反転させることで簡単に計算できます。

YOLOv3は、ロジスティック回帰を使用して各境界ボックスのオブジェクトネススコアを予測します。以前のバウンディングボックスが他の前のバウンディングボックスよりもグラウンドトゥルースオブジェクトと重なっている場合、これは1になります。以前のバウンディングボックスが最適ではないが、グラウンドトゥルースオブジェクトがしきい値を超えてオーバーラップしている場合は、[17]に従って予測を無視します。しきい値は5を使用します。 [17]とは異なり、システムでは、グラウンドトゥルースオブジェクトごとに1つの境界ボックスのみを割り当てます。以前のバウンディングボックスがグラウンドトゥルースオブジェクトに割り当てられていない場合、座標またはクラス予測の損失は発生せず、オブジェクト性のみが発生します。

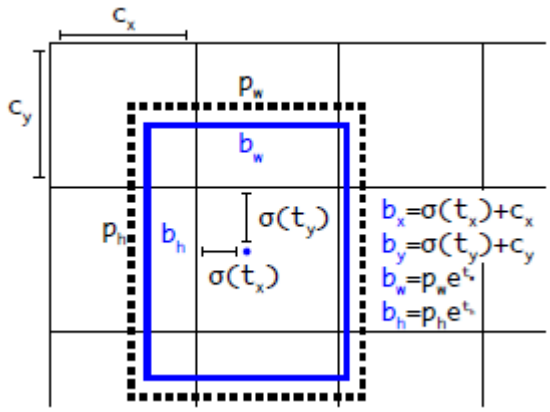

図2.寸法の事前分布と位置予測を含む境界ボックス。ボックスの幅と高さをクラスターの重心からのオフセットとして予測します。シグモイド関数を使用して、フィルターアプリケーションの位置を基準としたボックスの中心座標を予測します。この図は、[15]から露骨に自己盗用しました。

2.2. Class Prediction

各ボックスは、マルチラベル分類を使用して、境界ボックスに含まれるクラスを予測します。優れたパフォーマンスには不要であることがわかったので、softmaxは使用しません。代わりに、独立したロジスティック分類子を使用します。トレーニング中に、クラス予測にバイナリクロスエントロピー損失を使用します。

この定式化は、Open Images Dataset [7]のようなより複雑なドメインに移行するときに役立ちます。このデータセットには、多くの重複するラベル（つまり、女性と人物）があります。 softmaxを使用すると、各ボックスにクラスが1つしかないという仮定が課せられますが、多くの場合そうではありません。マルチラベルアプローチは、データをより適切にモデル化します。

2.2. Predictions Acrross Scales

YOLOv3は、3つの異なるスケールでボックスを予測します。私たちのシステムは、ピラミッドネットワークを特徴とする同様の概念を使用して、これらのスケールから特徴を抽出します[8]。基本特徴抽出器から、いくつかの畳み込み層を追加します。これらの最後は、3次元テンソルエンコーディングの境界ボックス、オブジェクト性、およびクラスを予測します。 COCO [10]の実験では、各スケールで3つのボックスを予測するため、4つのバウンディングボックスオフセット、1つの客観性予測、80のクラス予測のテンソルは$N \times N \times [3 \ast（4 + 1 + 80)]$です。

次に、前の2つのレイヤーから特徴マップを取得し、$2 \times$でアップサンプリングします。また、ネットワークの初期の特徴マップを取得し、連結を使用してそれをアップサンプリングされたフィーチャとマージします。この方法により、アップサンプリングされた機能からより意味のある情報を取得し、以前の特徴マップからより詳細な情報を取得できます。次に、さらにいくつかの畳み込み層を追加して、この結合された特徴マップを処理し、最終的には2倍のサイズになりますが、同様のテンソルを予測します。

同じ設計をもう一度実行して、最終的なスケールのボックスを予測します。したがって、3番目のスケールの予測は、以前のすべての計算と、ネットワークの早い段階からのきめ細かい機能の恩恵を受けます。

引き続き、バウンディングボックスの事前分布を決定するためにk平均クラスタリングを使用します。 9つのクラスターと3つのスケールを任意に選択し、クラスターをスケール全体で均等に分割するだけです。 COCOデータセットでは、9つのクラスターは$(10×13), (16×30), (33×23), (30×61), (62×45), (59×119), (116×90), (156×198), (373×326)$。

2.4. Feature Extractor

特徴抽出を実行するために新しいネットワークを使用します。私たちの新しいネットワークは、YOLOv2、Darknet-19で使用されているネットワークと、その新しくできた残余ネットワークとの間のハイブリッドアプローチです。私たちのネットワークは連続する$3×3$と$1×1$のたたみ込み層を使用していますが、いくつかのショートカット接続もあり、非常に大きくなっています。 53の畳み込み層があるので、それを呼び出します$...$待ってください$...$ Darknet-53！

	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
	Avgpool		Global	
	Connected		1000	
	Softmax			

表1. Darknet-53.

各ネットワークは同一の設定でトレーニングされ、$256×256$、単一クロップ精度でテストされます。実行時間は、$256×256$のTitan Xで測定されます。したがって、Darknet-53は、最先端の分類子と同等のパフォーマンスを発揮しますが、浮動小数点演算が少なく、速度が向上します。 Darknet-53はResNet-101よりも優れており、1.5倍高速です。 Darknet-53はResNet-152と同様のパフォーマンスを持ち、2倍高速です。

Darknet-53は、1秒あたりの測定された最高の浮動小数点演算も実現します。これは、ネットワーク構造がGPUをより効率的に利用することを意味し、評価がより効率的になり、したがってより高速になります。これは主に、ResNetのレイヤー数が多すぎて効率が悪いためです。

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

表2.バックボーンの比較。精度、数十億回の操作、毎秒10億回の浮動小数点操作、さまざまなネットワークのFPS。

2.5. Training

ハードネガティブマイニングなどを一切行わずに、完全な画像をトレーニングします。マルチスケールのトレーニング、大量のデータ拡張、バッチの正規化、すべての標準的なものを使用しています。トレーニングとテストには、Darknetニューラルネットワークフレームワークを使用します[14]。

3. How We Do

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

表3.これらすべてのテーブルを真剣に盗んでいるだけです[9]。
ゼロから作成するには非常に時間がかかります。はい、YOLOv3は問題なく動作しています。
RetinaNetは、画像の処理に$3.8\times$ほど長くかかることに注意してください。 YOLOv3はSSDバリアントよりもはるかに優れており、AP50メトリックの最新モデルに匹敵します。

YOLOv3はかなり良いです表3を参照してください。COCOに関しては、奇妙な平均AP距離がSSDバリアントと同等ですが、3倍高速です。ただし、この距離では、RetinaNetなどの他のモデルよりもかなり遅れています。

しかし、$IOU = .5$（またはグラフのAP_{50}）でのmAPの「古い」検出メトリックを見ると、YOLOv3は非常に強力です。 RetinaNetとほぼ同等で、SSDバリアントをはるかに上回っています。これは、YOLOv3がオブジェクト用のまともなボックスの作成に優れた非常に強力な検出器であることを示しています。ただし、IOUしきい値が増加すると、YOLOv3がボックスをオブジェクトと完全に整列させるのに苦労しているため、パフォーマンスは大幅に低下します。

過去には、YOLOは小さな物体に苦しんでいました。しかし、今ではその傾向が逆転しています。新しいマルチスケール予測では、YOLOv3のAPSパフォーマンスが比較的高いことがわかります。ただし、中規模および大きなサイズのオブジェクトでは、パフォーマンスが比較的低下します。この問題の原因を突き止めるには、さらに調査が必要です。

精度と速度をAP_{50}メトリックでプロットすると（図5を参照）、YOLOv3が他の検出システムよりも優れていることがわかります。つまり、より高速で優れています。

4. Things We Tried Thea Didn't Work

YOLOv3に取り組んでいる間、私たちはたくさんのことを試しました。それの多くはうまくいきませんでした。ここに私たちが覚えることができるものがあります。

アンカーボックス x,y オフセット予測。 xを予測する通常のアンカーボックス予測メカニズムを使用してみました。線形アクティブ化を使用したボックスの幅または高さの倍数としてのyオフセット。この定式化によりモデルの安定性が低下し、うまく機能しないことがわかりました。

線形x,y ロジスティックの代わりにy予測。線形活性化を使用してxを直接予測してみました。ロジスティック活動化の代わりにyオフセット。これにより、mAPが2ポイント低下しました。

焦点の喪失。焦点損失を使用してみました。 mAPが約2ポイント低下しました。 YOLOv3は、オブジェクティブ予測と条件付きクラス予測が別々であるため、フォーカルロスが解決しようとしている問題に対してすでに堅牢である可能性があります。したがって、ほとんどの例では、クラス予測からの損失はありませんか？か何か？完全にはわかりません。

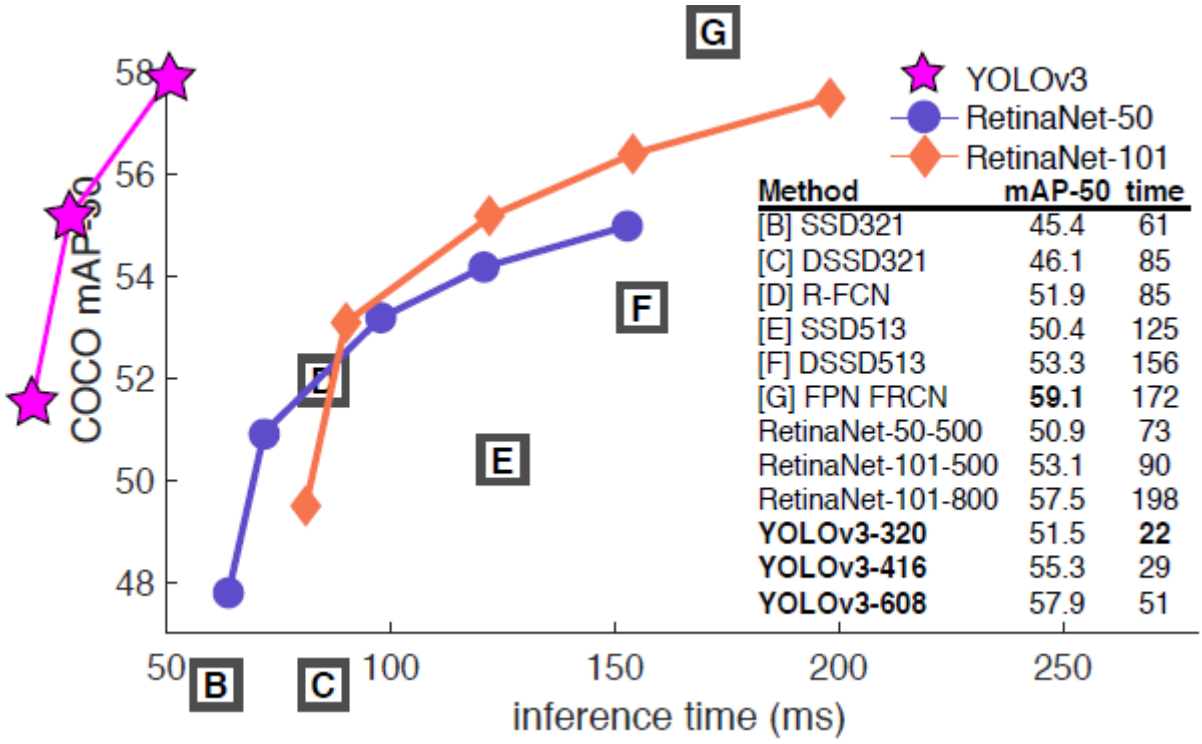

図3.再度[9]から改作したもので、今回は.5 IOUメトリックでのmAPの速度/精度のトレードオフを示しています。 YOLOv3は非常に高く、左端にあるため、良いと言えます。自分の論文を引用できますか？誰が試してみるか、この男！ [16]。ああ、忘れました、YOLOv2のデータ読み込みのバグも修正しました。レイアウトを崩さないように、ここに忍び込んでください。

5. What This All Means

YOLOv3は優れた検出器です。高速で正確です。 COCO平均APで0.5〜.95 IOUメトリックほど優れていません。しかし、これは.5 IOUの古い検出メトリックでは非常に優れています。

とにかくメトリックを切り替えたのはなぜですか？元のCOCOペーパーには、次のような不可解な文があります。「評価サーバーが完成したら、評価指標の完全な議論が追加されます」。 Russakovsky et alは、人間は.3と.5のIOUを区別するのに苦労していると報告しています！「IOU 0.3の境界ボックスを視覚的に検査し、IOU 0.5の境界ボックスと区別するように人間を訓練することは、驚くほど難しいことです。」 [18]人間が違いを伝えるのに苦労している場合、それはどれほど重要ですか？

しかし、おそらくより良い質問は、「これらの検出器を手に入れたので、これらの検出器をどうするか」です。この研究をしている多くの人々はグーグルとフェイスブックにいます。少なくとも私たちはテクノロジーがうまく機能していて、個人情報を収集してそれを販売するのに絶対に使用されないことを知っていると思います...待って、それはまさにそれが何のために使用されるのかということですか？ああ。

さて、ビジョン研究に多額の資金を提供している他の人々は軍隊であり、彼らは新しい技術で多くの人々を殺すような恐ろしいことをしたことは一度もありません.... 1

コンピュータービジョンを使用しているほとんどの人が、国立公園のシマウマの数を数えたり[13]、家の周りを歩いている猫を追跡したり[19]。しかし、コンピュータービジョンはすでに疑わしい使用に供されており、研究者として、私たちには少なくとも私たちの仕事がもたらす害を考慮し、それを軽減する方法を考える責任があります。私たちは世界にその分を借りています。

最後に、not @ meは使用しないでください。 （私がついにTwitterをやめたからです）。