YOLOv3: uma melhoria incremental

Joseph Redmon Ali Farhadi
universidade de Washington

Abstrato

Apresentamos algumas atualizações para o YOLO! Fizemos um monte de pequenas alterações de design para torná-lo melhor. Também treinamos essa nova rede que é muito boa. É um pouco maior do que da última vez, mas mais preciso. Ainda é rápido, não se preocupe. Em 320 × 320 YOLOv3 é executado em 22 ms a 28,2 mAP, tão preciso quanto o SSD, mas três vezes mais rápido. Quando olhamos para a antiga métrica de detecção de mAP de .5 IOU, YOLOv3 é muito boa. Ele atinge 57,9 AP50 em 51 ms em um Titan X, comparado a 57,5 AP50 em 198 ms por RetinaNet, desempenho semelhante, mas 3,8 vezes mais rápido. Como sempre, todo o código está online em https://pjreddie.com/yolo/.

1. Introdução

Às vezes você meio que liga por um ano, sabe? Eu não fiz muita pesquisa este ano. Passou muito tempo no Twitter. Brincou um pouco com GANs.
Eu tinha um pouco de impulso que sobrou do ano passado [12] [1]; Consegui fazer algumas melhorias no YOLO. Mas, honestamente, nada super interessante, apenas um monte de pequenas mudanças que o tornam melhor. Eu também ajudei um pouco com a pesquisa de outras pessoas.

Na verdade, é isso que nos traz aqui hoje. Temos um prazo para a câmera pronta [4] e precisamos citar algumas das atualizações aleatórias que fiz no YOLO, mas não temos uma fonte. Então prepare-se para um RELATÓRIO TÉCNICO!

O melhor dos relatórios de tecnologia é que eles não precisam de introduções, vocês sabem por que estamos aqui. Assim, o final desta introdução servirá de guia para o restante do artigo. Primeiro vamos dizer qual é o negócio com YOLOv3. Então vamos dizer-lhe como fazemos. Também falaremos sobre algumas coisas que tentamos e que não funcionaram. Finalmente vamos contemplar o que tudo isso significa.

2. O Acordo

Então aqui está o acordo com o YOLOv3: Na maioria das vezes, pegamos boas ideias de outras pessoas. Também treinamos uma nova rede classificadora que é melhor que as outras. Vamos levá-lo através de todo o sistema do zero para que você possa entender tudo.

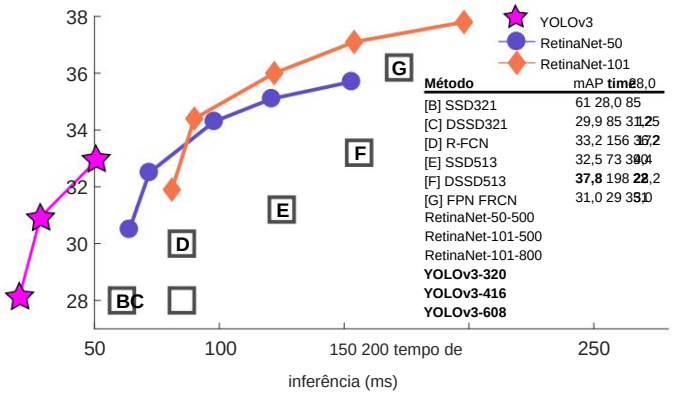

Figura 1. Adaptamos esta figura do artigo Focal Loss [9]. O YOLOv3 é executado significativamente mais rápido do que outros métodos de detecção com desempenho comparável. Vezes de um M40 ou Titan X, eles são basicamente a mesma GPU.

2.1. Previsão de caixa delimitadora

Seguindo o YOLO9000, nosso sistema prevê caixas delimitadoras usando clusters de dimensão como caixas âncora [15]. A rede prevê 4 coordenadas para cada caixa delimitadora, tx, ty, tw, th. Se a célula estiver deslocada do canto superior esquerdo da imagem por (cx, cy) e a caixa delimitadora anterior tiver largura e altura pw, ph, as previsões correspondem a:

bx = ÿ(tx) + cx por
= ÿ(ty) + cy
tw bw = pwe
th bh = phe

Durante o treinamento, usamos a soma da perda de erro ao quadrado. Se o verdade de base para alguma previsão de coordenadas é * nossa gra t^dient é o valor de verdade de base (calculado a partir da caixa de verdade de base) menos nossa previsão: ŷ * Este valor pode ser facilmente calculado invertendo as equações acima.

YOLOv3 prevê uma pontuação de objetividade para cada caixa delimitadora usando regressão logística. Deve ser 1 se a caixa delimitadora anterior se sobrepuser a um objeto de verdade mais do que qualquer outra caixa delimitadora anterior. Se a caixa delimitadora anterior

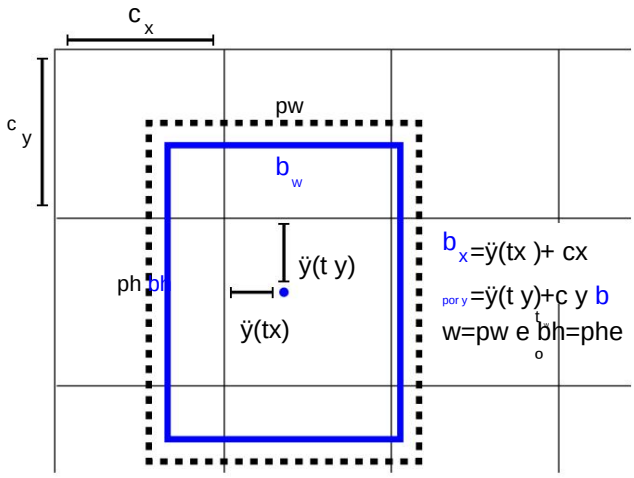

Figura 2. **Caixas delimitadoras com priorização de dimensão e previsão de localização.** Prevemos a largura e a altura da caixa como deslocamentos dos centróides do cluster. Prevemos as coordenadas do centro da caixa em relação ao local de aplicação do filtro usando uma função sigmóide. Esta figura descaradamente auto-plágio de [15].

não é o melhor, mas se sobrepõe a um objeto de verdade do terreno por mais do que algum limite, ignoramos a previsão, seguindo [17]. Usamos o limite de 0,5. Ao contrário de [17] , nosso sistema atribui apenas uma caixa delimitadora anterior para cada objeto de verdade. Se uma caixa delimitadora anterior não for atribuída a um objeto de verdade, ela não incorre em perda para predições de coordenadas ou classes, apenas objetividade.

2.2. Previsão de classe

Cada caixa prevê as classes que a caixa delimitadora pode conter usando classificação multirrótulo. Nós não usamos um softmax porque descobrimos que é desnecessário para um bom desempenho, em vez disso, simplesmente usamos classificadores logísticos independentes. Durante o treinamento, usamos perda de entropia cruzada binária para as previsões de classe.

Esta formulação ajuda quando passamos para domínios mais complexos como o Open Images Dataset [7]. Neste conjunto de dados existem muitos rótulos sobrepostos (ou seja, Mulher e Pessoa). O uso de um softmax impõe a suposição de que cada caixa possui exatamente uma classe, o que geralmente não é o caso. Uma abordagem multilabel modela melhor os dados.

2.3. Previsões em todas as escalas

YOLOv3 prevê caixas em 3 escalas diferentes. Nosso sistema extrai características dessas escalas usando um conceito similar para caracterizar redes de pirâmide [8]. Do nosso extrator de recursos básicos, adicionamos várias camadas convolucionais. A última delas prevê uma caixa delimitadora de codificação de tensor 3-d, objetividade e previsões de classe. Em nossos experimentos com COCO [10] prevemos 3 caixas em cada escala, então o tensor é N × N × [3 ÿ (4 + 1 + 80)] para os 4 deslocamentos de caixa delimitadora, 1 previsão de objetividade e 80 previsões de classe.

Em seguida, pegamos o mapa de recursos de 2 camadas anteriores e o aumentamos em 2×. Também pegamos um mapa de recursos de anteriormente na rede e o mesclamos com nossos recursos de upsampled usando concatenação. Esse método nos permite obter informações semânticas mais significativas dos recursos de upsampled e informações mais refinadas do mapa de recursos anterior. Em seguida, adicionamos mais algumas camadas convolucionais para processar esse mapa de recursos combinado e, eventualmente, prevemos um tensor semelhante, embora agora com o dobro do tamanho.

Executamos o mesmo projeto mais uma vez para prever caixas para a escala final. Assim, nossas previsões para a 3ª escala se beneficiam de toda a computação anterior, bem como de recursos refinados desde o início da rede.

Ainda usamos o agrupamento k-means para determinar nossas prioris de caixa delimitadora. Nós apenas escolhemos 9 clusters e 3 escalas arbitrariamente e depois dividimos os clusters igualmente entre as escalas. No conjunto de dados COCO, os 9 clusters foram: (10×13), (16×30),(33×23),(30×61),(62×45),(59×119),(116×90) ,(156 × 198),(373 × 326).

2.4. Extrator de recursos

Usamos uma nova rede para realizar a extração de recursos. Nossa nova rede é uma abordagem híbrida entre a rede usada no YOLOv2, Darknet-19 e esse material de rede residual ultramoderno. Nossa rede usa sucessivas camadas convolucionais 3 × 3 e 1 × 1, mas agora também possui algumas conexões de atalho e é significativamente maior. Ele tem 53 camadas convolucionais, então nós o chamamos.... espere... Darknet-53!

	Modelo	Filtros	Tamanho	Saída
	Convolucional	32		256 × 256
	Convolucional	64	Tamanho 3 × 3 / 2	128 × 128
1×	Convolucional	32	1 × 1	
	Convolucional	64	3 × 3	
	Residual			128 × 128
	Convolucional	128	3 × 3 / 2	64 × 64
2×	Convolucional	64	1 × 1	
	Convolucional	128	3 × 3	
	Residual			64 × 64
	Convolucional	256	3 × 3 / 2	32 × 32
8×	Convolucional	128	1 × 1	
	Convolucional	256	3 × 3	
	Residual			32 × 32
	Convolucional	512	3 × 3 / 2	16 × 16
8×	Convolucional	256	1 × 1	
	Convolucional	512	3 × 3	
	Residual			16 × 16
	Convolucional	1024	3 × 3 / 2	8 × 8
4×	Convolucional	512	1 × 1	
	Convolucional	1024	3 × 3	
	Residual			8 × 8
	AvgpoolGenericName		Global	
	Conectado		1000	
	Softmax			

Tabela 1. **Darknet-53.**

Essa nova rede é muito mais poderosa que a Darknet 19, mas ainda mais eficiente que a ResNet-101 ou ResNet-152.

Aqui estão alguns resultados do ImageNet:

Espinha dorsal	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	**93,8**	1039	53
ResNet-152 [5]	**77,6**	29,4			1090	37
Darknet-53	77,2	**93,8**	18,7		**1457**	78

Tabela 2. **Comparação de backbones.** Precisão, bilhões de operações, bilhões de operações de ponto flutuante por segundo e FPS para diversas redes.

Cada rede é treinada com configurações idênticas e testada a 256×256, precisão de corte único. Os tempos de execução são medidos em um Titan X em 256 × 256. Assim, o Darknet-53 funciona em par com classificadores de última geração, mas com menos flutuante operações pontuais e mais velocidade. Darknet-53 é melhor que ResNet-101 e 1,5x mais rápido. Darknet-53 tem desempenho semelhante ao ResNet-152 e é 2x mais rápido.

Darknet-53 também atinge a flutuação medida mais alta operações pontuais por segundo. Isso significa que a estrutura de rede utiliza melhor a GPU, tornando a avaliação mais eficiente e, portanto, mais rápida. Isso ocorre principalmente porque as ResNets têm apenas muitas camadas e não são muito eficientes.

2.5. Treinamento

Ainda treinamos em imagens completas sem mineração negativa difícil ou qualquer uma dessas coisas. Usamos treinamento em várias escalas, muitos dados aumento, normalização em lote, todas as coisas padrão. Usamos a estrutura de rede neural Darknet para treinamento e testes [14].

3. Como fazemos

YOLOv3 é muito bom! Veja a tabela 3. Em termos de COCOs métrica média de AP média estranha está no mesmo nível do SSD variantes, mas é 3x mais rápido. Ainda está um pouco atrás de outros modelos como RetinaNet nesta métrica.

No entanto, quando olhamos para a métrica de detecção “antiga” de mAP em IOU= .5 (ou AP50 no gráfico) YOLOv3 é muito Forte. Está quase no mesmo nível do RetinaNet e muito acima as variantes SSD. Isso indica que YOLOv3 é um detector forte que se destaca na produção de caixas decentes para objetos. No entanto, o desempenho cai significativamente à medida que o IOU o limite aumenta, indicando que o YOLOv3 se esforça para obter o caixas perfeitamente alinhadas com o objeto.

No passado YOLO lutava com pequenos objetos. No entanto, agora vemos uma reversão nessa tendência. Com o novo previsões em várias escalas, vemos que o YOLOv3 tem Desempenho APS . No entanto, tem comparativamente pior desempenho em objetos de tamanho médio e grande. Mais investigação é necessária para chegar ao fundo disso.

Quando plotamos precisão versus velocidade na métrica AP50 (consulte figura 5) vemos que o YOLOv3 tem benefícios significativos em relação a outros sistemas de detecção. Ou seja, é mais rápido e melhor.

4. Coisas que tentamos que não funcionaram

Tentamos muitas coisas enquanto trabalhávamos YOLOv3. Muito não funcionou. Aqui estão as coisas que podemos lembrar.

Caixa âncora x, y **previsões de deslocamento.** Tentamos usar o mecanismo de previsão de caixa âncora normal onde você prevê o deslocamento x, y como um múltiplo da largura ou altura da caixa usando uma ativação linear. Descobrimos que esta formulação diminuiu a estabilidade do modelo e não funcionou muito bem.

Previsões **lineares** x, y **em vez de logísticas.** Nós tentamos usando uma ativação linear para prever diretamente o deslocamento x, y em vez da ativação logística. Isso levou a um ponto de casal queda no mAP.

Perda focal. Tentamos usar a perda focal. Derrubou nosso mAP cerca de 2 pontos. YOLOv3 pode já ser robusto para o problema que a perda focal está tentando resolver porque tem previsões de objetividade separadas e previsões de classe condicional. Assim, para a maioria dos exemplos, não há perda do previsões de classe? Ou alguma coisa? Não temos certeza.

	espinha dorsal	AP	AP50	AP75	APS	APM	APL
Métodos de dois estágios							
R-CNN+++ mais rápido [5] 34,9 R-CNN mais rápido w FPN [8] 36,2 R-CNN mais rápido por G-RMI [6] Inception-ResNet-v2 [21] 34,7 R-CNN mais rápido w TDM [20] Inception-ResNet -v2-TDM 36.8 Métodos de um estágio	ResNet-101-C4	55,7	37,4	15,6	38,7	50,9
	ResNet-101-FPN	59,1	39,0	18,2	39,0	48,2
		55,5	36,7	13,5	38,1	52,0
		57,7	39,2	16,2	39,8	**52.1**
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

Tabela 3. Estou seriamente roubando todas essas tabelas do [9] que demoram muito para serem feitas do zero. Ok, YOLOv3 está indo bem. Tenha em mente que o RetinaNet tem 3,8 vezes mais tempo para processar uma imagem. YOLOv3 é muito melhor do que as variantes SSD e comparável ao modelos de última geração na métrica AP50 .

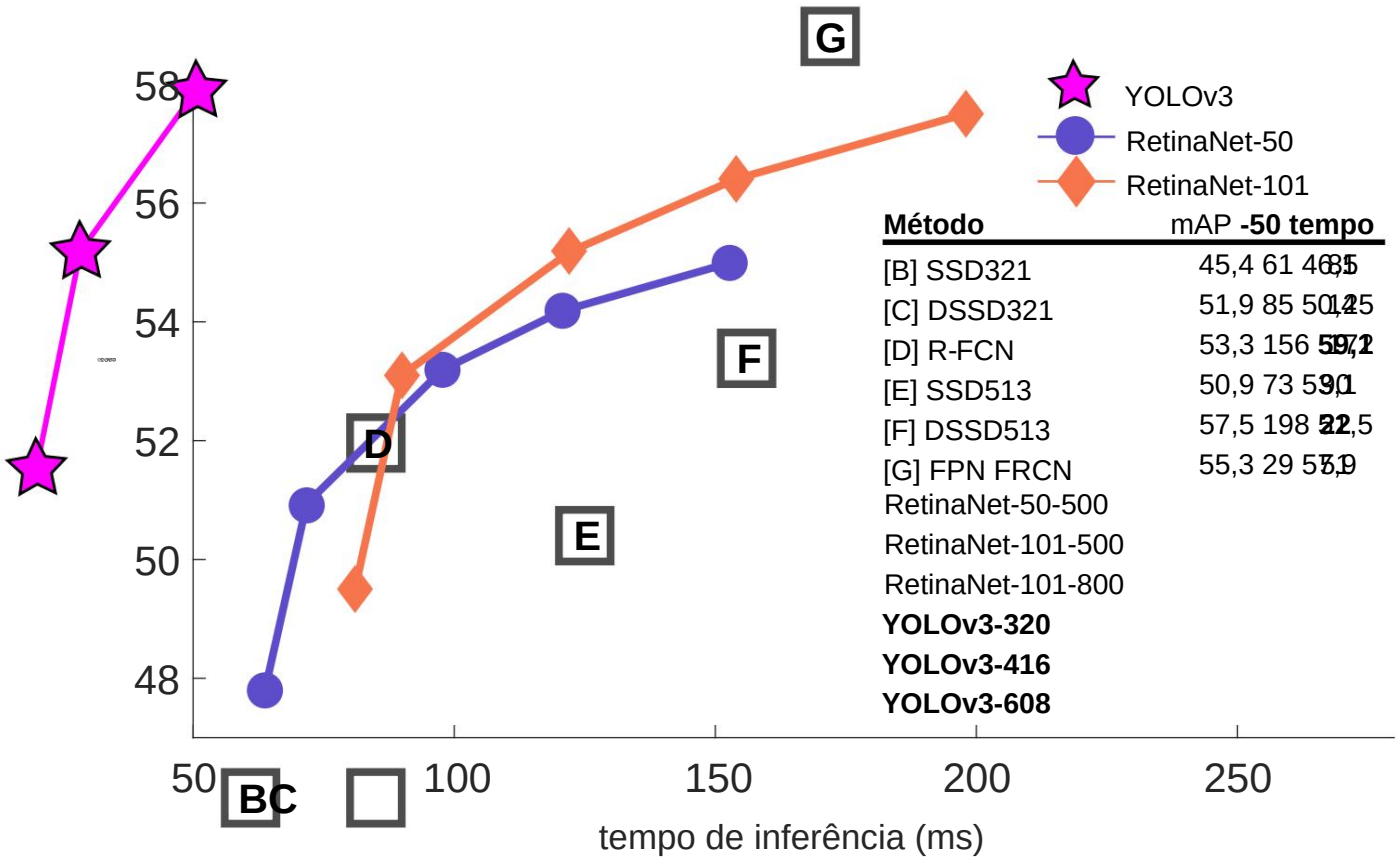

Figura 3. Novamente adaptado de [9], desta vez exibindo a relação velocidade/precisão no mAP na métrica 0,5 IOU. Você pode dizer que o YOLOv3 é bom porque é muito alto e muito à esquerda. Você pode citar seu próprio trabalho? Adivinha quem vai tentar, esse cara ÿ [16]. Ah, esqueci, também corrigimos um bug de carregamento de dados no YOLOv2, que ajudou em 2 mAP. Apenas colocando isso aqui para não atrapalhar o layout.

Limites de IOU duplos e atribuição de verdade. Faster R CNN usa dois limites de IOU durante o treinamento. Se uma predição se sobrepuser à verdade básica por .7 é como um exemplo positivo, por [.3ÿ.7] é ignorado, menos de .3 para todos os objetos de verdade básica é um exemplo negativo. Tentamos uma estratégia semelhante, mas não conseguimos bons resultados.

Gostamos bastante de nossa formulação atual, parece estar em um ótimo local, pelo menos. É possível que algumas dessas técnicas possam eventualmente produzir bons resultados, talvez apenas precisem de algum ajuste para estabilizar o treinamento.

5. O que tudo isso significa

YOLOv3 é um bom detector. É rápido, é preciso. Não é tão bom no AP médio COCO entre 0,5 e 0,95 métrica IOU. Mas é muito bom na antiga métrica de detecção de 0,5 IOU.

Por que mudamos as métricas de qualquer maneira? O documento original do COCO tem apenas esta frase enigmática: “Uma discussão completa das métricas de avaliação será adicionada assim que o servidor de avaliação estiver completo”. Russakovsky et al relatam que os humanos têm dificuldade em distinguir um IOU de 0,3 de 0,5! “Treinar humanos para inspecionar visualmente uma caixa delimitadora com IOU de 0,3 e distingui-la de uma com IOU de 0,5 é sur surpreendentemente difícil.” [18] Se os humanos têm dificuldade em dizer a diferença, o quanto isso importa?

Mas talvez uma pergunta melhor seja: “O que vamos fazer com esses detectores agora que os temos?” Muitas das pessoas que fazem essa pesquisa estão no Google e no Facebook.
Acho que pelo menos sabemos que a tecnologia está em boas mãos e definitivamente não será usada para coletar suas informações pessoais e vendê-las para... espere, você está dizendo que é exatamente para isso que ela será usada? Oh.

Bem, as outras pessoas que financiam pesadamente a pesquisa da visão são os militares e eles nunca fizeram nada horrível como matar muitas pessoas com novas tecnologias, oh espere 1

Tenho muita esperança de que a maioria das pessoas que usam a visão computacional esteja apenas fazendo coisas boas e felizes com ela, como contar o número de zebras em um parque nacional [13], ou rastrear seu gato enquanto ele vagueia pela casa [19]. Mas a visão computacional já está sendo colocada em uso questionável e, como pesquisadores, temos a responsabilidade de pelo menos considerar o dano que nosso trabalho pode estar causando e pensar em maneiras de mitigá-lo. Devemos isso ao mundo.

Para encerrar, não @ me. (Porque eu finalmente saí do Twitter).

1O autor é financiado pelo Office of Naval Research e pelo Google.

Referências

[1] Analogia. Wikipedia, março de 2018. 1

[2] M. Everingham, L. Van Gool, CK Williams, J. Winn e A. Zisserman. O desafio de classes de objetos visuais (voc) pascal. International journal of computer vision, 88(2):303–338, 2010. 6 [3] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi e AC Berg.

Dssd: Detector deconvolucional de disparo único. arXiv pré-impressão arXiv:1701.06659, 2017. 3

[4] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox e A. Farhadi. Iqa: Resposta visual de perguntas em ambientes interativos. arXiv pré-impressão arXiv:1712.03316, 2017. 1 [5] K. He, X. Zhang, S. Ren e J. Sun. Aprendizado residual profundo para reconhecimento de imagem. Em Anais da conferência IEEE sobre visão computacional e reconhecimento de padrões, páginas 770–778, 2016. 3

[6] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al.
Compensações de velocidade/precisão para detectores de objetos convolucionais modernos. 3

[7] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie , V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan e K. Murphy. Imagens abertas: um conjunto de dados público para classificação de imagens multirrótulo e multiclasse em larga escala. Conjunto de dados disponível em https://github.com/openimages, 2017. 2

[8] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan e S. Belongie. Redes de pirâmide de recursos para detecção de objetos. Em Anais da Conferência IEEE sobre Visão Computacional e Reconhecimento de Padrões, páginas 2117–2125, 2017. 2, 3 [9] T.-Y. Lin, P. Goyal, R. Girshick, K. He e P. Dollar. Perda focal para detecção de objetos densos. arXiv pré-impressão arXiv:1708.02002, 2017. 1, 3, 4

[10] T.-Y. Lin, M. Maire,.S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar e CL Zitnick. Microsoft coco: Objetos comuns em contexto. Na conferência europeia sobre visão computacional, páginas 740–755. Springer, 2014. 2 [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu e AC Berg. Ssd: Detector multibox de disparo único. Na conferência europeia sobre visão computacional, páginas 21–37. Springer, 2016. 3

[12] I. Newton. Philosophiae naturalis principia mathematica. William Dawson & Sons Ltd., Londres, 1687. 1 [13] J. Parham, J. Crall, C. Stewart, T. Berger-Wolf e D. Rubenstein. Censamento da população animal em escala com ciência cidadã e identificação fotográfica. 2017. 4 [14] J. Redmon. Darknet: Redes neurais de código aberto em c. http://pjreddie.com/darknet/, 2013-2016. 3 [15] J. Redmon e A. Farhadi. Yolo9000: Melhor, mais rápido, mais forte. Em Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, páginas 6517–6525. IEEE, 2017. 1, 2, 3 [16] J. Redmon e A. Farhadi. Yolov3: Uma melhoria incremental. arXiv, 2018. 4

[17] S. Ren, K. He, R. Girshick e J. Sun. R-cnn mais rápido: para detecção de objetos em tempo real com redes de proposta de região. arXiv pré-impressão arXiv:1506.01497, 2015. 2

[18] O. Russakovsky, L.-J. Li e L. Fei-Fei. O melhor dos dois mundos: colaboração homem-máquina para anotação de objetos. Em Anais da Conferência IEEE sobre Visão Computacional e Reconhecimento de Padrões, páginas 2121–2131, 2015. 4

[19] M. Scott. Smart camera gimbal bot scanlime: 027, dezembro de 2017. 4

[20] A. Shrivastava, R. Sukthankar, J. Malik e A. Gupta. Além de pular conexões: Modulação de cima para baixo para detecção de objetos. arXiv pré-impressão arXiv:1612.06851, 2016. 3

[21] C. Szegedy, S. Ioffe, V. Vanhoucke e AA Alemi. Inception-v4, inception-resnet e o impacto das conexões residuais na aprendizagem. 2017. 3

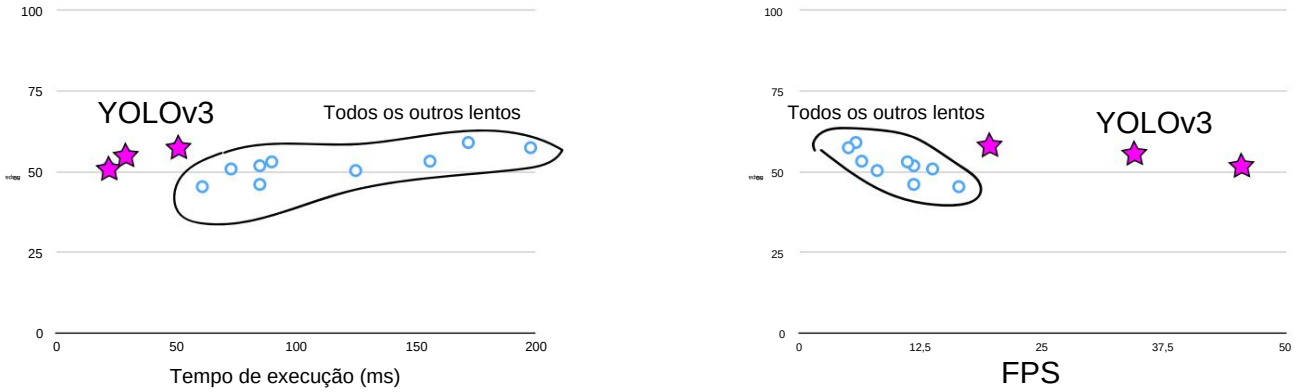

Figura 4. Gráficos de eixo zero são provavelmente mais honestos intelectualmente... e ainda podemos mexer com as variáveis para parecermos bons!

Refutação

Gostaríamos de agradecer aos comentaristas do Reddit, colegas de laboratório, e-mails e gritos no corredor por suas palavras adoráveis e sentidas no coração. Se você, como eu, está revisando para o ICCV, então sabemos que você provavelmente tem 37 outros artigos que poderia estar lendo que invariavelmente adiará até a última semana e, em seguida, terá alguma legenda no e-mail de campo sobre como você realmente deve terminar essas revisões, exceto que não ficará totalmente claro o que eles estão dizendo e talvez sejam do futuro? De qualquer forma, este papel não terá se tornado o que será com o tempo sem todo o trabalho que seus eus passados fizeram também no passado, mas apenas um pouco mais adiante, não como todo o caminho até agora. E se você twittou sobre isso, eu não saberia. Apenas dizendo.

O revisor nº 2, também conhecido como Dan Grossman (lol cegando quem faz isso) insiste que eu aponte aqui que nossos gráficos não têm uma, mas duas origens diferentes de zero. Você está absolutamente certo Dan, isso é porque parece muito melhor do que admitir para nós mesmos que estamos todos aqui lutando por 2-3% de mAP. Mas aqui estão os gráficos solicitados. Eu joguei um com FPS também porque parecemos super bons quando plotamos em FPS.

O revisor nº 4, também conhecido como JudasAdventus no Reddit, escreve “Entertain mas os argumentos contra as métricas do MSCOCO parecem um pouco fracos”. Bem, eu sempre soube que você seria o único a se voltar contra mim, Judas. Você sabe quando você trabalha em um projeto e só dá certo, então você tem que descobrir alguma maneira de justificar como o que você fez realmente foi muito legal? Eu estava basicamente tentando fazer isso e ataquei um pouco as métricas do COCO. Mas agora que estabeleci esta colina, posso morrer nela.

Veja, o mAP já está meio quebrado, então uma atualização para ele talvez deva resolver alguns dos problemas ou pelo menos justificar por que a versão atualizada é melhor de alguma forma. E essa é a grande coisa com a qual eu discordei foi a falta de justificativa. Para PASCAL VOC, o limite de IOU foi ”definido deliberadamente baixo para levar em conta imprecisões nas caixas delimitadoras nos dados de verdade” [2]. O COCO tem uma rotulagem melhor do que o VOC? Isso é definitivamente possível, pois o COCO tem máscaras de segmentação, talvez os rótulos sejam mais confiáveis e, portanto, não estamos tão preocupados com imprecisões. Mas, novamente, meu problema era a falta de justificativa.

A métrica COCO enfatiza melhores caixas delimitadoras, mas essa ênfase deve significar que ela não enfatiza outra coisa, neste caso, a precisão da classificação. Existe uma boa razão para pensar que mais caixas delimitadoras precisas são mais importantes do que uma melhor classificação? Um exemplo de classificação incorreta é muito mais óbvio do que uma caixa delimitadora levemente deslocada. O mAP já está estragado porque tudo o que importa é a ordem de classificação por classe. Por exemplo, se o seu conjunto de teste tiver apenas essas duas imagens, de acordo com o mAP, dois detectores que produzem esses resultados são TÃO BONS:

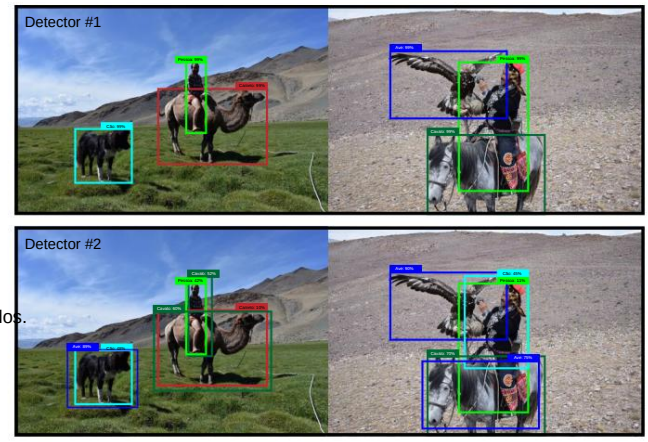

Figura 5. Esses dois detectores hipotéticos são perfeitos de acordo com o mAP sobre essas duas imagens. Ambos são perfeitos. Totalmente igual.

Agora, isso é OBVIAMENTE um exagero dos problemas com o mAP, mas acho que meu ponto recém-retransmitido é que existem discrepâncias tão óbvias entre o que as pessoas no “mundo real” se importariam e nossas métricas atuais que acho que se vamos apresentar novas métricas, devemos nos concentrar nessas discrepâncias. Além disso, tipo, já é a precisão média média, o que chamamos de métrica COCO, precisão média média média?

Aqui está uma proposta, o que as pessoas realmente se importam é dada uma imagem e um detector, quão bem o detector encontrará e classificará objetos na imagem. Que tal se livrar do AP por classe e apenas fazer uma precisão média global? Ou fazer um cálculo de AP por imagem e calcular a média?

As caixas são estúpidas de qualquer maneira, eu provavelmente sou um verdadeiro crente em máscaras, exceto que não consigo fazer com que YOLO as aprenda.