Comparação de Métodos de Deep Learning Pré-Treinados da Biblioteca OpenCV para Detecção de Pessoas em Ambientes Internos

Jesuino Vieira Filho[1], **Pablo Andretta Jaskowiak**[1]

[1]Centro Tecnológico de Joinville (CTJ)
Universidade Federal de Santa Catarina (UFSC)
Joinville, Santa Catarina – Brasil

jesuino.vieira@grad.ufsc.br,pablo.andretta@ufsc.br

Resumo. *Sistemas de monitoramento baseados em câmeras são cada vez mais onipresentes. A existência de um sistema de monitoramento não garante, porém, que todas as informações coletadas sejam utilizadas e/ou analisadas. Quando uma interpretação das imagens é necessária, usualmente recorre-se à visão computacional. Neste contexto particular, métodos de Deep Learning têm recebido crescente atenção. De fato, apesar de seu desenvolvimento recente, alguns destes métodos estão disponíveis em bibliotecas e pacotes de software de forma pré-treinada, permitindo sua aplicação com relativa facilidade. Neste trabalho diferentes modelos de Deep Learning disponíveis na biblioteca OpenCV foram comparados para a detecção e contagem de pessoas em ambientes internos. Os modelos foram comparados quanto à sua precisão, revocação e tempo de detecção. Os resultados obtidos sugerem que o método YOLOv3 apresenta um bom compromisso entre medida F_1 e tempo de reconhecimento.*

Abstract. *Camera based surveillance systems are ubiqutous nowadays. The existence of a monitoring system does not guarantee, however, that all collected images are analyzed. If interpretation of the images is required, computer vision methods are usually employed to automate the task. In this particular context, Deep Learning methods have received increasing attention. In fact, despite their recent development, some of these methods are available in pre-trained libraries and software packages, allowing their application with relative ease. In this work, different Deep Learning models available in the OpenCV library were compared for the detection and counting of people indoors. Models were compared w.r.t. precision, recall and detection time. Our results indicate that YOLOv3 has a good compromise between F_1 measure and detection time.*

1. Introdução

Sistemas de monitoramento eletrônico por câmeras são uma realidade atual e cada vez mais comum, tendo em vista a diminuição de seu custo de implantação. Os sistemas mais básicos disponíveis possuem funcionamento limitado à captura e armazenamento das imagens coletadas. Nestes sistemas, um operador humano se faz necessário para realizar uma análise posterior das imagens ou monitorá-las em tempo real. Além de demandar grande atenção e ser suscetível a erros, tal trabalho se mostra, muitas vezes, mais custoso do que a

implementação do próprio sistema [Collins et al. 2000]. A fim de diminuir o custo associado com a interpretação das imagens capturadas e obter sistemas mais reativos, isto é, sistemas que consigam emitir alertas ou interpretações em um curto espaço de tempo (ou até mesmo em tempo real), diversas técnicas computacionais têm sido aplicadas na literatura para a análise das imagens capturadas [Valera and Velastin 2005, Mainetti et al. 2014].

Do ponto de visa de análise, sistemas de visão computacional podem ser categorizados em sistemas de baixo e alto nível [Sonka et al. 2007]. Sistemas de baixo nível são caracterizados por considerar somente a imagem e por não utilizar informações semânticas, ou seja, não existe uma interpretação ou auxílio externo (informação externa) durante sua análise. Em contrapartida, sistemas de alto nível buscam adicionar semântica às cenas sob avaliação [Sonka et al. 2007], identificando e nomeando objetos de forma automática. Em geral, o sistema aprende como realizar a detecção ou identificação dos objetos a partir de um conjunto já previamente rotulado de imagens, que serve como base de treinamento. Dentre os métodos passíveis de aplicação nestes sistemas, destacam-se os de Aprendizado de Máquina [Mitchell 1997] e Deep Learning [Goodfellow et al. 2016].

Técnicas de Deep Learning têm recebido crescente atenção nos últimos anos, apresentando resultados excepcionais em aplicações de visão computacional [Guo et al. 2016]. A Biblioteca OpenCV [Bradski 2000], referência em visão computacional, disponibiliza alguns modelos pré-treinados destes métodos para a detecção de objetos particulares, como por exemplo, pessoas, pássaros e carros. É importante notar que a utilização de modelos pré-treinados não representa um cenário ideal de aplicação, podendo impactar negativamente os resultados observados. Entretanto, esta pode ser uma abordagem factível para prototipação ou até mesmo a única alternativa para alguns usuários e tem sido utilizada em trabalhos recentes [Chevtchenko et al. 2018]. Isto se deve ao fato de que a etapa de treinamento destes modelos é laboriosa, demandando a criação de bases de dados com um grande número de imagens rotuladas, usualmente obtidas a partir de anotações manuais das mesmas [Hinterstoisser et al. 2018].

Este trabalho tem por objetivo comparar modelos pré-treinados de Deep Learning, tal qual disponibilizados na biblioteca OpenCV (versão 3.4.3), para a detecção de pessoas em um ambiente interno. Mais especificamente, considerou-se um laboratório de pesquisa, ambiente no qual há uma baixa movimentação (em comparação à um corredor de shopping, por exemplo). Para realização dos experimentos computacionais uma base de dados de imagens foi criada considerando diferentes condições típicas de utilização. A base de dados obtida foi utilizada para aplicação dos modelos e comparação dos resultados com base em latência de detecção, precisão e revocação.

O restante do artigo está organizado como segue. Na Seção 2, é apresentada uma breve revisão acerca de técnicas e trabalhos relacionados à tarefa de detecção de pessoas. Na Seção 3, é apresentado o design experimental do trabalho, compreendendo uma discussão sucinta dos modelos comparados, coleta da base de dados e métricas de avaliação empregadas. Na Seção 4, os resultados obtidos a partir da comparação dos modelos são apresentados e discutidos. Na Seção 5, são apresentadas as conclusões do trabalho.

2. Abordagens para Detecção de Pessoas

A detecção de pessoas é uma tarefa fundamental e recorrente em visão computacional, porém considerada desafiadora [Topkaya et al. 2014]. Dificuldades surgem pois o corpo

humano pode apresentar-se em diferentes posições, ângulos de observação, ou ainda com oclusão. A detecção e contagem de pessoas baseada em vídeo possui vantagens como hardware simples, ausência de contato e riqueza de informações [Cai et al. 2014]. Os sistemas de contagem de pessoas tem uma ampla aplicação em estatísticas de fluxo de pessoas, otimização da programação de trabalho e, principalmente, na fiscalização de segurança [Antić et al. 2009]. No que se refere à ambientes internos estes sistemas podem ser utilizados para construção de ambientes inteligentes [Brumitt et al. 2000].

Em geral, o processo de detecção de pessoas a partir de imagens pode ser realizado nas seguintes etapas sequenciais [Nguyen et al. 2016]: (i) extração de regiões candidatas potencialmente cobertas por objetos humanos; (ii) descrição das regiões extraídas; (iii) classificação das regiões como humanas ou não humanas e; (iv) pós-processamento. Revisões acerca da tarefa de detecção de pessoas podem ser encontradas na literatura [Mainetti et al. 2014, Nguyen et al. 2016], demonstrando sua importância. Tais revisões também apontam que uma gama de métodos e abordagens têm sido empregadas para atacar o problema, como subtração de fundo [Zhang and Liang 2010], agrupamento de dados [Topkaya et al. 2014], Histograma de Gradientes Orientados (HOG) [Dalal and Triggs 2005] e técnicas de Aprendizado de Máquina, como Deep Learning [Chevtchenko et al. 2018].

É importante ressaltar que algumas abordagens não são capazes de realizar todas etapas anteriormente descritas. A subtração de fundo, por exemplo, pode auxiliar na geração de regiões candidatas mas não é capaz, por si só, de atribuir semântica às regiões obtidas. Abordagens baseadas em Deep Learning, foco deste trabalho, possuem a capacidade de construir internamente descritores (representações) para os objetos presentes nas imagens e de aprender características que não são observadas de forma explícita ou direta. Isto faz com que estes métodos sejam capazes de fazer a detecção de ponta a ponta, realizando todas as quatro etapas do processo de detecção anteriormente discutido.

Considerando métodos de Deep Learning, em particular, progressos significativos foram alcançados nos últimos anos [Huang et al. 2016]. A partir do bom desempenho alcançado pela rede AlexNet [Krizhevsky et al. 2012] na competição ImageNet 2012 [Russakovsky et al. 2014], outras abordagens de Deep Learning têm sido desenvolvidas e/ou empregadas para detecção de pessoas nos mais variados ambientes [Nguyen et al. 2016]. Considerando implementações prontamente acessíveis ao usuário final, como as redes pré-treinadas da biblioteca OpenCV [Bradski 2000], não há, até onde sabemos, trabalhos que abordem e/ou realizem uma comparação dos modelos, tais quais fornecidos, para detecção de pessoas em ambientes internos.

3. Materiais e Métodos

A seguir são discutidos os modelos comparados, a base de dados e critérios de avaliação.

3.1. Modelos

Neste trabalho, foram comparados modelos de detecção de objetos pré-treinados disponibilizados[1] pela biblioteca OpenCV – na sua versão 3.4.3. Geralmente, um detector de objeto moderno é composto de duas partes: um extrator de características e uma

[1] `https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/download_models.py`

cabeça [Bochkovskiy et al. 2020], representadas na Figura 1. O extrator de características é responsável por selecionar e combinar variáveis que descrevem a imagem em características (por exemplo, detecção de bordas), reduzindo a quantidade de dados que devem ser processados, enquanto ainda representa a imagem original. A cabeça é utilizada para prever as classes e localizar o objeto na imagem, baseado nas características extraídas no passo anterior. Quanto à parte da cabeça, esta geralmente é categorizada em dois tipos: detector de objetos de um estágio (previsão densa) e detector de objetos de dois estágios (previsão esparsa). Um sumário dos modelos utilizados é apresentado na Tabela 1. Optou-se por utilizar somente modelos devidamente documentados e diferenciá-los entre as duas categorias previamente descritas.

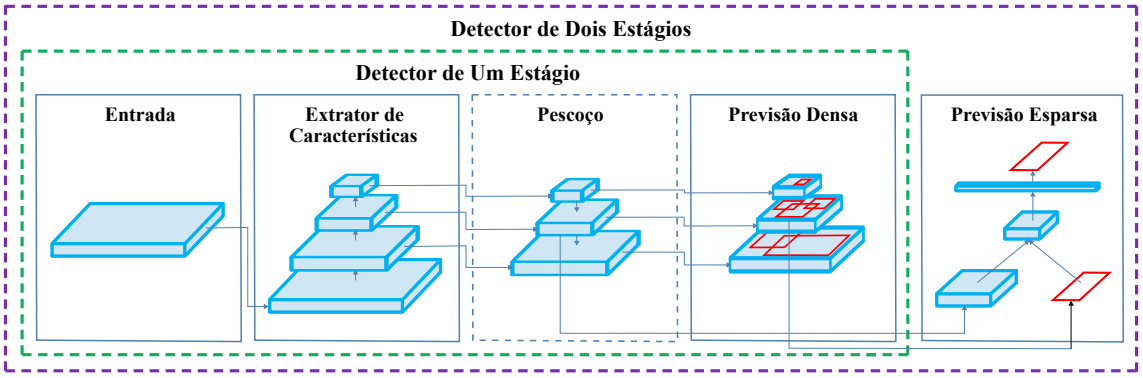

Figura 1. Detector de objetos. Adaptado de [Bochkovskiy et al. 2020].

Os detectores de dois estágios são assim chamados pois utilizam um algoritmo para propor regiões de interesse e um classificador para processar apenas as regiões selecionadas. Tendo como exemplo os métodos R-FCN [Dai et al. 2016], Faster R-CNN [Ren et al. 2015] e Mask R-CNN [He et al. 2017], uma rede de propostas regionais (RPN) é utilizada para a primeira tarefa. Em seguida, utilizando os recursos extraídos pelo extrator de características e as regiões de interesse, Faster R-CNN extrai os recursos que corresponderiam aos objetos relevantes em um novo tensor, o qual é finalmente classificado. Mask R-CNN é uma extensão do método anterior, com algumas mudanças nas camadas da rede e a adição da segmentação – isto é, também retornará a máscara do objeto. Utilizando uma abordagem diferente, R-FCN move a complexidade principal da tarefa de classificação para a geração de mapas de pontuação sensíveis à posição, propondo melhores regiões e demandando menos processamento na tarefa de classificação.

Os detectores de um estágio, por outro lado, aceleram o processo eliminando a necessidade da rede de proposta da região. No método YOLO [Redmon and Farhadi 2016, Redmon and Farhadi 2018], a imagem é dividida em uma grade SxS e, para cada célula desta grade, a posição do objeto e as probabilidades de classe (e.g., pessoa, carro, cachorro) são previstas diretamente com uma única rede. De modo similar, o método SSD [Liu et al. 2015] gera K caixas delimitadores de diferentes proporções, formas e escalas em cada local nos mapas de características extraídos no passo anterior. Para compensar a menor precisão decorrente de uma abordagem mais simples, estes métodos aplicam otimizações (como características de múltiplas escalas) que permitem que seus resultados sejam comparáveis à métodos de dois estágios, ainda que com maior velocidade.

Naturalmente, ambas abordagens tem a extração de características como

Tabela 1. Modelos comparados e suas principais características.

	Método	Extrator de Características	Framework	Resolução da Imagem	Base de Dados de Treinamento
1 Estágio	SSD	MobileNet	Caffe	300x300	COCO + VOC
	SSD	MobileNetV1	TensorFlow	300x300	COCO
	SSD	MobileNetV2	TensorFlow	300x300	COCO
	SSD	InceptionV2	TensorFlow	300x300	COCO
	YOLOv2	Darknet-19	Darknet	416x416	VOC
	YOLOv3	Darknet-53	Darknet	416x416	COCO
2 Estágios	R-FCN	ResNet-50	Caffe	800x600	VOC
	Faster R-CNN	ResNet-50	TensorFlow	800x600	COCO
	Faster R-CNN	InceptionV2	TensorFlow	800x600	COCO
	Mask R-CNN	InceptionV2	TensorFlow	1280x720	COCO

um componente central no *pipeline* de detecção, dado que os recursos extraídos deste processo alimentarão a cabeça do detector. Estes algoritmos utilizam redes neurais convolucionais e, no que diz respeito a abordagens para detecção de pessoas, têm como objetivo capturar a forma, aparência ou informações de movimento do objeto humano [Nguyen et al. 2016]. Dentre os extratores de características utilizados nos modelos comparados neste trabalho, estão: MobileNet [Howard et al. 2017], Inception [Szegedy et al. 2014], DarkNet [Redmon and Farhadi 2016, Redmon and Farhadi 2018] e ResNet [He et al. 2016]. Uma revisão mais detalhada de redes neurais convolucionais que servem como extratores de características em modelos de detecção de objetos é apresentada por [Benali Amjoud and Amrouch 2020].

É importante notar que os modelos utilizados são desenvolvidos em diferentes frameworks e, principalmente, treinados em diferentes base de dados. As base de dados Pascal VOC [Everingham et al. 2015] e/ou MS-COCO [Lin et al. 2014] foram utilizadas para treinar os modelos aqui comparados. Além disso, outros fatores influenciam a qualidade dos detectores, como as configurações utilizadas durante o treinamento (e.g., tamanho do lote e taxa de aprendizado) e diferentes resoluções de imagem de entrada. Neste ponto, é importante ressaltar que estamos comparando as implementações tal qual disponibilizadas e não o método em si, devido aos fatores citados. Todos os modelos já foram treinados para reconhecimento de pessoas. Para todos os experimentos foi considerado um valor limite de confiança de 0.5. Isto é, se o método possui uma confiança maior da presença de uma pessoa do que de sua ausência, indica-se a presença de uma pessoa.

3.2. Aquisição da Base de Dados

Para realizar a comparação dos modelos, uma base de dados real foi obtida a partir da captura de vídeos no laboratório. As imagens foram obtidas com uma câmera *HP Webcam HD-4110*. Para tanto, foram gravados sete vídeos, cada um com cerca de cinco minutos de duração. Os vídeos foram capturados na resolução de 1280x720. Cada um dos vídeos gravado considera um número fixo de pessoas no laboratório, variando de uma até sete pessoas. Os usuários do laboratório se movimentaram livremente no período de gravação,

gerando situações usuais de utilização do ambiente – inclusive com casos de oclusão. Para o processo de avaliação, foram extraídas 100 imagens (*frames*) de cada vídeo, totalizando 700 imagens. A fim de obter uma maior diversidade de situações, as imagens foram selecionadas em intervalos regulares em cada vídeo, abrangendo toda sua duração e ignorando imagens similares. A Figura 2 apresenta duas imagens (redimensionadas) da base de dados. Os rostos foram pixelados apenas para apresentação.

Figura 2. Duas imagens da base de dados com duas e três pessoas.

3.3. Medidas de Avaliação

O objetivo deste trabalho é avaliar os modelos na forma em que são disponibilizados pela biblioteca OpenCV. Desta forma, as afirmações e resultados obtidos dizem respeito as implementações disponibilizadas pela biblioteca OpenCV. Os modelos foram executados e avaliados considerando a configuração padrão. Diante disso, foram analisadas as seguintes métricas: precisão, revocação, medida F_1 [Goodfellow et al. 2016].

Para calcular o tempo de execução, foi considerado apenas o tempo que cada modelo leva para processar uma única imagem. Para obter uma melhor estimativa, o tempo de detecção para cada uma das 700 imagens foi computado individualmente. Reportamos, ao final, o valor médio e desvio padrão. Para execução de todos os *scripts* Python desenvolvidos, foi utilizado uma máquina com processador *Intel Core i5 2.30 GHz* e 8 GB de memória DRAM. É importante ressaltar que a execução de todos modelos ocorreu perante as mesmas condições: após ligar o computador, sem outros programa em execução e sem o auxílio de placa de vídeo. Todos modelos processaram os mesmos *frames*.

4. Resultados e Discussão

Os resultados obtidos a partir da execução dos modelos na base de dados anteriormente descrita são apresentados na Tabela 2. Os valores em cada célula da tabela correspondem às médias obtidas durante a avaliação no conjunto de 700 imagens. Valores de desvio padrão são apresentados entre parênteses. Para cada uma das medidas de avaliação adotadas, o modelo com melhor resultado é destacado em negrito.

Em suas configurações padrões, tais quais disponibilizadas pela OpenCV, os modelos apresentam, em geral, uma precisão alta. Considerando o critério precisão, o melhor resultado foi obtido pelo modelo YOLOv2/Darknet-19, que apresentou precisão igual a 1.00. Em outras palavras, o modelo nunca indicou a presença equivocada de uma pessoa (Falso Positivo). Os piores resultados de precisão são ainda consideravelmente altos.

Analisando os dados de revocação é possível perceber que, tal qual empregados, grande parte dos modelos são conservadores com relação à suas classificações (valores

Tabela 2. Resultados de precisão, revocação, medida F_1 e tempo de execução.

Modelo	Precisão	Revocação	$\mathbf{F_1}$	Tempo (s)
SSD / MobileNet	0.99 (0.02)	0.67 (0.28)	0.76 (0.25)	0.09 (0.01)
SSD / MobileNetV1	0.92 (0.13)	0.85 (0.19)	0.87 (0.16)	**0.06 (0.00)**
SSD / MobileNetV2	0.99 (0.02)	0.63 (0.27)	0.73 (0.24)	0.09 (0.02)
SSD / InceptionV2	0.99 (0.01)	0.71 (0.26)	0.80 (0.22)	0.12 (0.01)
YOLOv2 / Darknet-19	**1.00 (0.00)**	0.52 (0.30)	0.62 (0.29)	0.48 (0.06)
YOLOv3 / Darknet-53	0.99 (0.02)	0.92 (0.14)	0.95 (0.11)	0.96 (0.06)
R-FCN / ResNet-50	0.99 (0.01)	0.89 (0.16)	0.93 (0.11)	2.37 (0.05)
Faster R-CNN / ResNet-50	0.97 (0.07)	0.93 (0.12)	0.94 (0.09)	1.09 (0.05)
Faster R-CNN / InceptionV2	0.98 (0.05)	0.92 (0.13)	0.95 (0.09)	3.38 (0.07)
Mask R-CNN / InceptionV2	0.96 (0.08)	**0.97 (0.07)**	**0.96 (0.06)**	3.44 (0.08)

mais baixos de revocação). Isso indica que, por vezes, os modelos deixaram de identificar pessoas presentes nas imagens (aumento nos valores de Falso Negativo). O único modelo que apresentou precisão ótima (YOLOv2) de fato se mostra o mais conservador de todos os modelos analisados. Com exceção do modelo YOLOv3, todos os modelos de um estágio apresentaram revocações mais baixas, em comparação aos modelos de dois estágios. Além de uma menor complexidade, tais modelos trabalham também com uma menor resolução das imagens de entrada, o que pode, em parte, corroborar para tais resultados. O tempo requerido pelo modelo YOLOv3 é, porém, até 9 vezes maior do que os demais modelos de um estágio (exceto YOLOv2). Já os modelos de dois estágios apresentam valores mais altos de revocação, indicando uma melhor recuperação e identificação das pessoas presentes nas imagens. Os valores obtidos para estes modelos apresentam, em média, uma revocação superior a 89%.

Os resultados da medida F_1 apresentam de forma combinada os resultados de precisão e revocação. Neste contexto, o melhor compromisso entre precisão e revocação é apresentado pelo modelo Mask R-CNN/InceptionV2 (F_1 de 0.96). Neste sentido, o modelo não só apresenta baixas taxas de falsa detecção, como também uma alta taxa de recuperação das pessoas realmente presentes nas imagens. Ademais, os melhores resultados, em geral, foram obtidos pelos modelos de dois estágios, juntamente com o modelo YOLOv3, de um estágio. É importante notar que estes modelos possuem um maior custo computacional em relação aos demais e podem ser proibitivos em aplicações nas quais o poder computacional é limitado. Nestes casos, em particular, o modelo SSD / MobileNetv1 apresenta um bom compromisso entre valores de medida F_1 e tempo de execução.

É possível notar que o tempo de execução por imagem dos modelos de um único estágio (parte superior da tabela) é, em geral, menor quando comparado ao dos métodos de dois estágios. Além deste fator, as resoluções das imagens de entrada também devem ser levadas em conta. Este último fator provavelmente leva à um maior tempo de execução para os métodos YOLOv2 e YOLOv3. Os modelos de dois estágios demandaram mais tempo para o processamento das imagens, variando entre 1s e 3s. Considerando o uso de um computador de uso pessoal, os únicos modelos que se mostram adequados para uma aplicação de tempo real são os de um estágio. Naturalmente, com o uso de *hardware* dedicado os outros métodos também seriam passíveis de utilização. De fato, extratores de características como MobileNet foram desenvolvidos visando especificamente aplicações

móveis, o que justifica baixo tempo de execução [Howard et al. 2017].

Tendo em vista que os valores de revocação tiveram uma maior variação do que os de precisão, nesta seção avaliamos a distribuição do número de pessoas que deixaram de ser identificadas em cada imagem, ou seja, quão grande foi a subestimação do número de pessoas em cada imagem avaliada. Para tanto, apresentamos os resultados de revocação considerando cada uma das 700 imagens avaliadas no *Boxplot* da Figura 3. Os modelos de dois estágios, juntamente com YOLOv3 (de um único estágio), apresentaram uma melhor estimativa por imagem. Dentre os modelos de um estágio, o modelo SSD / MobileNetV1 apresenta o melhor perfil de estimação de número de pessoas por imagem. De fato, seu perfil é próximo (ainda que pior) dos observados com os modelos de dois estágios.

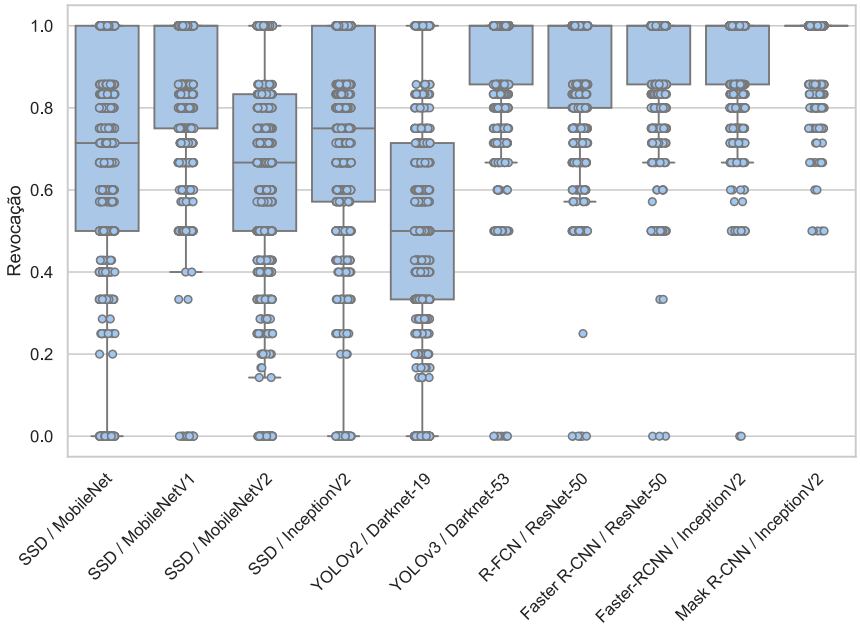

Figura 3. *Boxplot* com a distribuição de revocação dos modelos para cada uma das 700 imagens avaliadas. Cada ponto é o resultado de uma imagem.

5. Conclusões

Embora a comparação realizada tenha utilizado modelos pré-treinados (aplicados com sua configuração padrão), os resultados indicam que é possível obter bons níveis de detecção. Um dos fatores que pode corroborar para esta observação é o fato do ambiente de aplicação ser relativamente "bem comportado", isto é, não há aglomerações, mudanças bruscas de luminosidade ou muitos fatores confundidores. Em geral, o modelo Mask R-CNN / InceptionV2 apresentou os melhores resultados considerando precisão e revocação. Seu tempo de processamento por imagem é, porém, da ordem de 3s em um computador de uso pessoal. Dentre os modelos avaliados, YOLOv3 / Darknet-53 apresentou um dos melhores compromissos entre precisão, revocação e tempo de execução.

Agradecimentos

Os autores agradecem ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e aos revisores, por suas críticas construtivas no processo de revisão.

Referências

Antić, B., Letić, D., Ćulibrk, D., and Crnojević, V. (2009). K-means based segmentation for real-time zenithal people counting. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2565–2568.

Benali Amjoud, A. and Amrouch, M. (2020). Convolutional neural networks backbones for object detection. In El Moataz, A., Mammass, D., Mansouri, A., and Nouboud, F., editors, *Image and Signal Processing*, pages 282–289, Cham. Springer International Publishing.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Brumitt, B., Meyers, B., Krumm, J., Kern, A., and Shafer, S. (2000). Easyliving: Technologies for intelligent environments. In *International Symposium on Handheld and Ubiquitous Computing*, pages 12–29. Springer.

Cai, Z., Yu, Z. L., Liu, H., and Zhang, K. (2014). Counting people in crowded scenes by video analyzing. In *2014 9th IEEE Conference on Industrial Electronics and Applications*, pages 1841–1845.

Chevtchenko, S., Vale, R., Cordeiro, F., and Macario, V. (2018). Deep learning for people detection on beach images. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 218–223.

Collins, R., Lipton, A., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D., Enomoto, N., and Hasegawa, O. (2000). A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Carnegie Mellon University, Pittsburgh, PA.

Dai, J., Li, Y., He, K., and Sun, J. (2016). R-FCN: object detection via region-based fully convolutional networks. *CoRR*, abs/1605.06409.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 886–893, Washington, DC, USA. IEEE Computer Society.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT Press.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187:27 – 48. Recent Developments on Deep Big Vision.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. *CoRR*, abs/1703.06870.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hinterstoisser, S., Lepetit, V., Wohlhart, P., and Konolige, K. (2018). On pre-trained image features and synthetic images for deep learning. In *The European Conference on Computer Vision (ECCV) Workshops*.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861.

Huang, J., Rathod, V., Sun, C., Zhu, M., Balan, A. K., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K. (2016). Speed/accuracy trade-offs for modern convolutional object detectors. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3296–3297.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., and Berg, A. C. (2015). SSD: single shot multibox detector. *CoRR*, abs/1512.02325.

Mainetti, L., Patrono, L., and Sergi, I. (2014). A survey on indoor positioning systems. In *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 111–120.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Nguyen, D. T., Li, W., and Ogunbona, P. O. (2016). Human detection from images and videos: A survey. *Pattern Recognition*, 51:148 – 175.

Redmon, J. and Farhadi, A. (2016). YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *CoRR*, abs/1804.02767.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. (2014). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252.

Sonka, M., Hlavac, V., and Boyle, R. (2007). *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CoRR*, abs/1409.4842.

Topkaya, I. S., Erdogan, H., and Porikli, F. (2014). Counting people by clustering person detector outputs. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 313–318.

Topkaya, I. S., Erdogan, H., and Porikli, F. (2014). Counting people by clustering person detector outputs. In *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 313–318.

Valera, M. and Velastin, S. A. (2005). Intelligent distributed surveillance systems: a review. *IEE Proceedings - Vision, Image and Signal Processing*, 152(2):192–204.

Zhang, L. and Liang, Y. (2010). Motion human detection based on background subtraction. In *2010 Second International Workshop on Education Technology and Computer Science*, volume 1, pages 284–287.