Real-time Ball Detection for Robocup Soccer Using Convolutional Neural Networks

Lucas Ribeiro de Abreu
Laboratório de I.A. Aplicada à Automação e Robótica
Centro Universitário da FEI
São Bernardo, Brazil
lucasribeiroabreu@gmail.com

Reinaldo Augusto da Costa Bianchi
Laboratório de I.A. Aplicada à Automação e Robótica
Centro Universitário da FEI
São Bernardo, Brazil
rbianchi@fei.edu.br

***Resumo*—The RoboCup Soccer is one of the largest competitions in the robotics field of research. It considers the soccer match as a challenge for the robots and aims to win a match between humans *versus* robots by the year of 2050. The vision module is a critical system for the robots because it needs to quickly locate and classify objects of interest for the robot in order to generate the next best action. In this paper, an approach using Convolutional Neural Networks for object detection is described. The soccer ball is the chosen object and three state-of-art convolutional neural networks architectures were trained for the experiment using data augmentation and transfer learning techniques. The models were evaluated in a test set, yielding promising results in precision and frames per second. The best model achieved an average precision of 0.972 with an intersection over union of 50% and 9.64 frames per second, running on CPU.**

***Index Terms*—RoboCup, Object Detection, Convolutional Neural Network, MobileNetV2, Faster R-CNN.**

I. Introdução

A *RoboCup* é uma competição anual científica destinada a promover a pesquisa no ramo da Robótica e da Inteligência Artificial. Ela considera o futebol como uma referência atraente ao público em geral, além de ser dinâmica, competitiva e cooperativa para testar novas tecnologias no estado da arte. A *RoboCup* possui diversas ligas, incluindo ligas de futebol por simulação computacional, ligas de futebol de robôs físicos, ligas de tarefas de resgate, domésticas e industriais. Quase todas as ligas envolvem agentes robôs que devem agir de forma autônoma em seus ambientes. A longo prazo, o objetivo é construir um time de futebol de robôs capaz de derrotar o time campeão mundial até 2050 [1], cumprindo as regras oficiais da FIFA.

A visão computacional é um dos componentes que devem ter um bom desempenho para que os robôs sejam capazes de realizar suas atividades ao longo da partida de futebol. O sistema deve classificar determinados objetos a partir de câmeras e fornecer sua posição exata em tempo real. Muitos algoritmos de aprendizado de máquina foram aplicados com sucesso nas últimas décadas para a detecção de objetos a partir de técnicas clássicas como HOG (*Histogram of Oriented Gradients*) em conjunto com máquina de vetores de suporte [2] e *boosted cascade* [3]. Além de outros métodos que ganharam muita popularidade nos últimos anos, como as redes neurais convolucionais, (CNN do inglês *Convolutional Neural network* ou *ConvNet*).

Redes neurais convolucionais tornaram-se o principal meio de visão computacional desde que a arquitetura AlexNet [6] popularizou redes neurais convolucionais profundas ao vencer o *ImageNet Challenge*: ILSVRC 2012 [4]. A partir de 2012, o ajuste de arquiteturas neurais profundas para obter a máxima precisão equilibrada com um bom desempenho tem sido uma área de pesquisa bastante ativa. Tanto a pesquisa manual de arquitetura quanto as melhorias nos algoritmos de treinamento levaram melhorias significativas em relação aos projetos iniciais, como a VGGNet [5], a GoogLeNet [7] e a ResNet [8].

As redes neurais revolucionaram muitas áreas de inteligência de artificial, permitindo precisão sobre-humana para tarefas desafiadoras de reconhecimento de imagem. No entanto, o impulso para melhorar a precisão muitas vezes tem um custo: redes de última geração exigem recursos computacionais elevados, os quais estão além das capacidades de muitos dispositivos móveis e embarcados [9].

A evolução constante da visão dos robôs é requerida, por exemplo, porque até 2014 a cor bola de futebol na RoboCup Soccer era completamente laranja, mas a partir de 2015 as especificações mudaram para uma bola com pelo menos 50% de cor branca deixando o resto da bola aberta para qualquer combinação de cores. Essas mudanças na complexidade afetaram negativamente os resultados de muitos algoritmos que eram utilizados, porque simples algoritmos de segmentação de cores combinado com identificação de formas não são efetivos com este tipo de bola [10]. A Fig. 1 ilustra a aparência contrastante da bola em três imagens coletadas em um campo de futebol de robôs. Foi aplicada uma amplificação nas imagens para melhor visualização. Nelas, é possível observar diferentes iluminações na bola, além de orientações e movimentos que comprometem a detecção.

A detecção de objetos é um assunto recorrente nas publicações anuais RoboCup [10] [11], inclusive sendo selecionado como melhor artigo na categoria de Contribuição à Engenharia na *RoboCup 2016: Robot World Cup XX* [10]. O prêmio de Contribuição à Engenharia na *RoboCup 2017: Robot World Cup XXI* também foi relacionado ao tema de detecção de objetos utilizando CNNs, mas o objeto em questão

(a) Bola de futebol estática e com a iluminação criando sombras em seu entorno.

(b) Bola apresenta distorção devido ao movimento.

(c) Imagem sofre com o brilho da iluminação e também contém a marca do pênalti, que pode ser confundida com a bola por conta de sua cor e formato.

Figura 1: Exemplos da aparência da bola em diferentes situações presenciadas pelo robô. Nota-se que as cores do gramado também apresentam variação de cor.

eram outros robôs presentes no campo de futebol [12].

Este artigo apresenta a aplicação de diferentes arquiteturas de rede neural ao contexto do futebol de robôs para a localização da bola de futebol. As redes aqui apresentadas são estado da arte para modelos de visão computacional em conjuntos de dados *benchmark*, considerando um balanço entre precisão e performance. Duas das redes testadas são adaptadas especificamente para dispositivos móveis, diminuindo significativamente o número de operações e a memória necessária, mantendo a precisão em níveis similares aos modelos com alto custo computacional. Os modelos serão detalhados nas próximas seções.

A principal contribuição é uma avaliação de arquiteturas robustas e inovadoras para ser embarcadas em robôs. Esta avaliação é feita a partir da porcentagem de precisão na detecção do objeto utilizando o conceito de IoU (interseção sobre a união, do inglês *Intersection over Union*), que é um conceito padrão na literatura quando se trata de detecção de objetos [5] [9] [13] [14] [15] [17], mas está ausente em artigos relevantes da *RoboCup* [10-12].

II. Trabalhos relacionados

O tema de localização de bola de futebol é recorrente nos anais da *Robocup World Cup*. Os autores do artigo [10] que ganhou o prêmio de Contribuição à Engenharia na *RoboCup 2016: Robot World Cup XX* utilizaram redes neurais convolucionais para localização da bola de futebol. As imagens foram imagens coletadas em laboratório, que possui um campo de jogo de futebol configurado de acordo as regras da liga Humanóide da *RoboCup*, mas reduzidas em tamanho. A motivação do estudo foi por conta da regra de coloração da bola que permite que 50% da superfície da bola seja de qualquer cor ou padrão, enquanto o restante deve permanece branco. Bolas multicoloridas possuem histogramas e padrões de cores variáveis dependendo da orientação e movimento, o que dificulta a identificação. Para atacar este desafio com uma abordagem inovadora [10], foram desenvolvidas duas arquiteturas de redes neurais convolucionais, treinadas com 1160 imagens, com o objetivo de gerar a previsão das coordenadas x e y relativas a uma distribuição normal da posição da bola no respectivo eixo. Desta forma, com a interseção das distribuições dos eixos, pode-se gerar uma área mais provável da bola estar localizada, conforme exemplifica a Fig. 2.

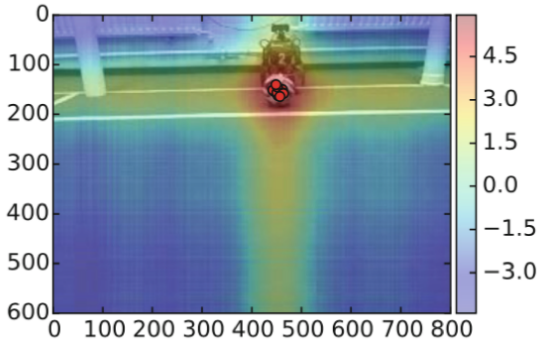


Figura 2: Exemplo do mapa de probabilidades de localização da bola. Fonte: [10] adaptado.

Apesar de inovadora e com precisão de 80% nos dados de teste com variações de mais ou menos 10 *pixels* para imagens complexas, os autores [10] concluem que é necessário criar um conjunto de dados maior e mais diverso para que seja possível obter resultados mais acurados. Além disso, o poder computacional dos robôs foi um empecilho para criar redes mais profundas e mais acuradas.

Para atacar o problema do volume de dados, em um artigo no contexto de detecção de bolas [13], porém de jogadores de *handball*, pesquisadores realizaram o treinamento de seis variações de uma arquitetura de CNN com apenas 1837 imagens. Fez-se o uso de transferência de conhecimento a partir de uma rede pré-treinada em um conjunto de dados que possui bolas de diversos esportes como parte das classes a serem detectadas. Além disso, o autor utilizou uma técnica de aumento de dados que consiste em gerar variações artificiais da imagem, que se mostrou muito efetiva pois economiza tempo na anotação de imagens e gera mais dados para a rede neural aprender. Como trabalho futuro, sugerem realizar mais transformações nos dados de treinamento para gerar mais variedade, pois foram as redes com esse tipo de operação que obtiveram a melhor performance. Ainda neste trabalho, foi

utilizado o conceito de IoU para avaliar as predições, pois garante que a precisão seja baseada em uma porcentagem específica de interseção entre a meta e o que foi predito.

Para resolver problemas de desempenho, existe uma classe de modelos eficientes chamados *MobileNets* que tem sido bastante difundida para aplicativos de visão móveis e integrados. As *MobileNets* baseiam-se em uma arquitetura simplificada que usa convoluções separáveis em profundidade para construir redes neurais profundas leves para realizar detecção de objetos, reconhecimento facial, classificação de imagens, entre outros. Experimentos [19] mostram que a rede atingiu resultados similares a outras redes no estado da arte mais profundas com apenas 1% do total de operações de multiplicação e adição.

Outra classe de modelo que possui ótimos resultados em *benchmarks* são as *Faster R-CNN* [16]. Métodos como este utilizam propostas de regiões para primeiro gerar potenciais detecções em uma imagem e, em seguida, executar um classificador nessas detecções propostas. Após a classificação, um pós-processamento é usado para refinar as caixas delimitadoras das detecções, eliminar as detecções duplicadas e recodificar as caixas com base em outros objetos na cena [15]. Porém, esses pipelines podem ser complexos, lentos e difíceis de otimizar, porque cada componente individual deve ser treinado separadamente, segundo estudos [14].

III. Metodologia

Nesta seção, serão apresentados os métodos, técnicas e banco de imagens utilizados para a realização do experimento. Tanto o banco de imagens quanto o código e arquivos de configuração estão disponíveis no *GitHub* [23].

A. Transferência de aprendizado

Treinar uma rede neural profunda com pesos inicializados de forma aleatória pode levar horas ou até dias. Uma alternativa é encontrar uma rede neural existente que realize uma tarefa semelhante àquela que se está tentando resolver [22]. A partir desta rede já treinada, é possível restaurar os parâmetros e pesos da rede neural na nova rede. Essa é uma forma de reutilizar os pesos das camadas e é chamado de transferência de aprendizado. Isso não só acelera consideravelmente o treinamento, mas também permite o uso de bases de dados reduzidas [17].

A transferência de aprendizado é utilizada principalmente para imagens, e possui algumas boas práticas para gerar bons resultados [17]:

- Descartar a camada de saída do modelo original, pois provavelmente não é útil para a nova tarefa, e pode não ter o número certo de saídas para a nova tarefa;
- Criar uma nova camada de saída com a quantidade certa de classes a serem treinadas;
- Congelar as camadas da rede, exceto a camada de saída. Ou seja, tornar os pesos não treináveis, para que a descida de gradiente modifique apenas a forma de interpretar os sinais recebidos das camadas de convolução para mapear as novas classes - e treinar o modelo com retro-propagação;
- Após algumas rodadas de treino, reduzir a taxa de aprendizado e descongelar uma ou duas das camadas ocultas superiores para permitir que a retro-propagação faça ajustes finos;

Como base, os pesos de uma rede pré-treinada no conjunto de dados COCO (*Common Objects in Context*) [26] foram utilizados nos três modelos deste artigo. O conjunto de dados COCO incorpora um grande número de classes, dentre elas, a classe de bolas esportivas, a qual contém bolas de futebol, tênis, *baseball*, entre outras. Portanto, a rede já possui pesos e a habilidade inerente de identificar objetos redondos, em diversos ambientes, iluminações e tamanhos.

B. Aumento de dados

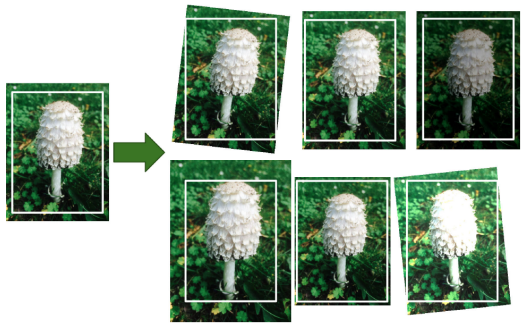


Figura 3: Exemplo de aumento de dados aplicado a imagens utilizando de operações de corte aleatório, rotação, translação, alteração de brilho. As técnicas também podem ser combinadas. [17].

Além da transferência de aprendizado, também foi utilizado o aumento de dados (do inglês: *data augmentation*). O principal objetivo é de gerar artificialmente mais imagens para o conjunto de dados de treinamento a partir da criação de variantes realistas de cada imagem de treinamento. Isso reduz o *overfitting*, tornando isso uma técnica de regularização [17]. As instâncias geradas devem ser as mais realistas possíveis, a ponto de um ser humano não ser capaz de dizer se foi aumentado ou não. Além disso, a simples inclusão de ruído branco não ajuda porque o ruído branco não deve ser aprendido por técnicas de aprendizado de máquina.

Exemplos de aumento de dados são operações de deslocar, girar e redimensionar levemente todas as imagens do conjunto de treinamento e adicionar as imagens resultantes ao conjunto de treinamento, conforme mostrado na Fig. 3. Isso força o modelo a ser mais tolerante a variações na posição, orientação e tamanho dos objetos nas imagens. Para aumentar a tolerância do modelo a diferentes condições de iluminação, pode-se gerar imagens com diferentes contrastes. Em geral, pode-se pode inverter as imagens horizontalmente (exceto para texto e outros objetos não simétricos) [17]. Ao combinar essas transformações, aumenta-se muito a variedade e volume dos de dados de treinamento.

C. Conjunto de dados

O experimento deste artigo utilizou uma base de dados contendo 3061 imagens de uma bola de futebol em miniatura em um campo de futebol de robôs, confome mostrado na Fig. 4. Todas as imagens foram anotadas, ou seja, tiveram marcação das caixas delimitadoras da bola de futebol. Elas foram criadas com o software gratuito *LabelImg*. O software salva as imagens em arquivos XML no formato PASCAL VOC [27], o formato utilizado no *ImageNet Challenge*.

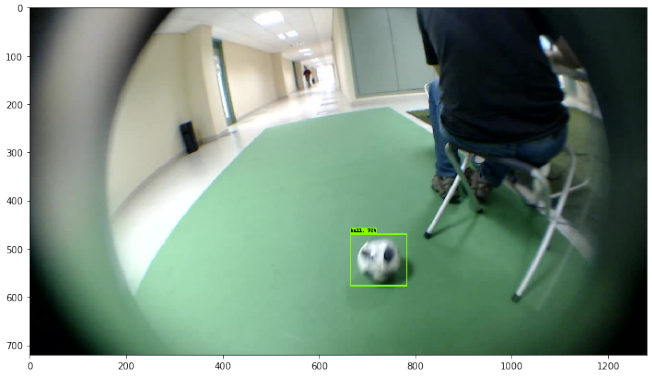


Figura 4: Exemplo de uma imagem de da base de dados para treinamento da localização da bola de futebol. Os dados anotados se referem às coordenadas da caixa delimitadora da bola. Fonte: Autor.

Este conjunto de dados foi separado em 80% treino (2449 imagens), 20% teste (612 imagens) e foi utilizado para o treinamento e avaliação dos modelos de redes neural apresentados a seguir. Em cima dos dados de treino, foram aplicadas técnicas de aumento de dados, conforme explicado na seção anterior.

D. Modelos

Conforme estudado nos trabalhos relacionados, a classe das *MobileNets* e as *Faster R-CNN* se adequam ao contexto de detecção de objetos. As *MobileNets* mais indicadas a sistemas embarcados e com uma boa quantidade de *frames* por segundo. E a *Faster R-CNN* se destacando por estar em um patamar do estado da arte em tarefas de detecção de objetos. O experimento consistirá em treinar estas redes utilizando as técnicas de aumento de dados e transferência de aprendizado a partir da rede COCO, que já possui treinamento prévio em classes relacionadas a bolas esportivas e será refinada para bolas de futebol.

1) MobileNetV2 e MobileNetV2Quantized: *MobileNets* baseiam-se em uma arquitetura simplificada que usa convoluções separáveis em profundidade. Esse tipo de convolução diminui significativamente a quantidade de operações realizadas para as previsões, tornando as redes mais leves e indicadas para dispositivos móveis [19].

Uma técnica para acelerar os modelos de redes convolucionais é a Quantificação [20]. Ambos os filtros em camadas convolucionais e matrizes de camadas totalmente conectadas são quantificados, visando minimizar o erro de estimativa da resposta de cada camada. Experimentos extensivos sobre o *benchmark ILSVRC-12* demonstram um fator de quatro a seis vezes de aceleração e uma compressão de quinze a vinte vezes com apenas um ponto percentual de perda na precisão de classificação. Baseado nisso, decidiu-se treinar tanto a *MobileNetV2* quanto a *MobileNetV2Quantized* para a tarefa de localização da bola.

Utilizou-se o otimizador *RMS Prop Optimizer* com ambos decaimento e momento definidos como 0,9. O tamanho dos lotes de treinamento eram de 12 imagens por aproximadamente 15 mil passos de treinamento.

2) Faster-RCNN: Outra arquitetura escolhida para o experimento foi a *Faster-R CNN*. O otimizador para esta rede foi o *Momentum Optimizer* com momento definido como 0,9. O tamanho dos lotes de treinamento eram de 8 imagens por aproximadamente 15 mil passos.

E. Métrica

Uma métrica muito comum usada em tarefas de detecção de objetos é a média de Precisão Média (AP) [17]. Para calcular a AP, calcula-se a precisão máxima que pode-se obter com pelo menos 0% de recall, 10% de recall, 20% e assim por diante até 100% e, em seguida, calcular a média dessas precisões máximas. Isso é chamado de métrica de Precisão Média. A (mAP) é a média entre as AP das classes sendo consideradas.

Para calcular o AP, o resultado da detecção do modelo é comparado com a caixa delimitadora verdadeira, considerando a IoU [13], que é ilustrada na Fig. 5 e representada pela Eq. 1. Pode-se medir a precisão com IoU maior ou igual a 50% ou 75%, por exemplo.

$$IoU = \frac{ÁreaInterseção}{ÁreaUnião} \tag{1}$$

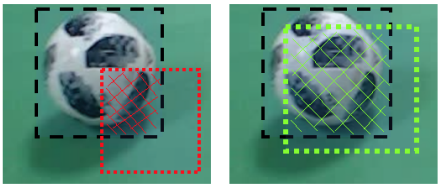


Figura 5: Exemplo IoU de caixa delimitadora verdadeira e detectada com menos de 50% IoU à esquerda e mais de 50% IoU à direita. Fonte: Autor.

IV. Resultados e Discussão

Para treinamento dos modelos, a plataforma de desenvolvimento de modelos de aprendizado profundo chamada *Tensorflow* [24] foi utilizada. Dentro desta plataforma, existe um módulo para Detecção de Objetos [25], onde existem os modelos previamente apresentados já treinados no conjunto de dados COCO.

As redes tomam como entrada uma imagem e retornam a localização do objeto, neste caso, a bola. E para acompanhar a evolução do aprendizado desta tarefa, a cada 150 passos de treinamento, avaliou-se com dados que não foram usados no treino, os chamados dados de teste, para que a evolução do desempenho possa ser observada graficamente. A Fig. 6 mostra essa evolução com os resultados *mAP* com 50% de IoU para o conjunto de teste.

É possível observar que a *MobileNetV2Quantized* obteve resultados superiores no início, porém as outras duas ultrapassaram a sua performance por volta dos 4 mil *steps*. A *Faster-RCNN* demonstrou menores variações ao longo do treinamento, enquanto as *MobileNets* apresentaram altos ganhos e perdas de precisão. Ao final, os resultados mAP com 50% IoU do último passo foram 0,923; 0,972; 0,891 para a *Faster R-CNN*, *MobileNetV2* e *MobileNetV2Quantized*, respectivamente.

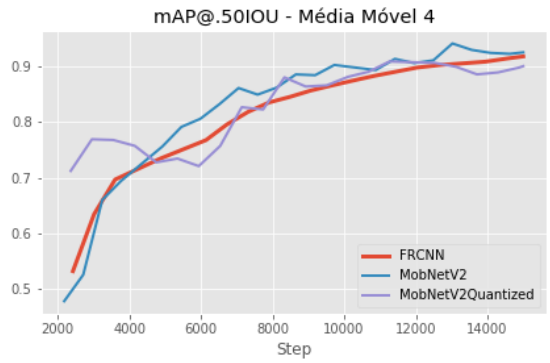


Figura 6: Resultados mAP com 50% de IoU para os passos de treinamento para cada um dos modelos. Foi aplicada uma média móvel de quatro passos para melhor visualização.

Já a Fig. 7 ilustra os resultados mAP 75% IoU, ou seja, representam resultados para detecções de objetos com maior área de interseção sobre a união das caixas delimitadoras reais e anotadas, o que é uma tarefa mais difícil. Por conta disso a precisão fica em patamares mais baixos. O modelo *Faster-RCNN* obteve ampla vantagem sobre os outros modelos durante todos os passos de treinamento. Os resultados finais foram: *Faster-RCNN* com 0,577; MobileNetV2 com 0,522; e MobileNetV2Quantized com 0,387.

Em ambos gráficos de mAP das Fig. 6 e Fig. 7 à 50% e à 75%, pode-se as redes *MobileNet* estavam atingindo um patamar que não indicava que haveria evolução na precisão, por conta de estarem se adaptando muito bem aos dados de treino e não generalizando bem para as imagens de teste, o chamado *overfitting*. As linhas estavam se estabilizando em torno de 90% de precisão com 50% IoU e 45% à 75% IoU. Porém a *Faster-RCNN*, aparentemente não mostra sinais de estagnação e poderia ser treinada por mais tempo. Por restrições de tempo máximo de uso da GPU e por metodologia, foi-se estipulado o estudo comparativo para 15 mil passos entre as três redes neurais. Para trabalhos futuros, a avaliação do ponto de máxima precisão, sem *overfitting*, do treinamento das

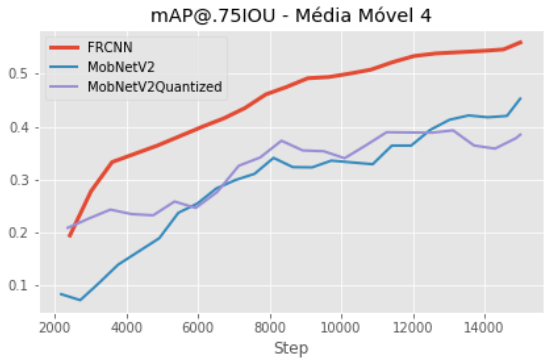
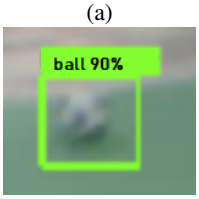


Figura 7: Resultados mAP com 75% de IoU por passos de treinamento. Foi aplicada uma média móvel de quatro passos para melhor visualização.

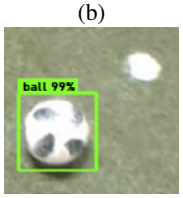
redes seria de grande valia, com uma infraestrutura de GPUs apropriada.

Na Fig. 8 pode-se observar alguns resultados de previsão dos modelos treinados. Para melhor visualização, foi aplicada uma amplificação nas imagens. Pode-se observar que os modelos ficaram robustos o bastante para identificar a bola com distorções de foco e a variações devido ao movimento, à distância e à luminosidade. Também foi possível observar nas imagens de teste que a marca do pênalti não foi confundida com a bola.

(a)



(b)



(c)

Figura 8: Exemplo aparência da bola dentro das caixas delimitadoras em algumas imagens de teste. Os modelos ficaram robustos o bastante para identificar a bola em movimento, conforme Subfig. 8a, com pouco foco, conforme Subfig. 8b e evitar falsos positivos quando imagem contém a marca do pênalti, conforme Subfig. 8c.

Apesar da *Faster-RCNN* prover melhores resultados com

IoU à 75% e resultados promissores à 50%, a rede claramente perde em medidas de performance em CPU (do inglês *Central Process Unit*), como a de *frames* por segundo (FPS) mostrado na Tabela I. As *MobileNets* obtiveram um fator de aproximadamente treze vezes a quantidade de imagens processadas por segundo. Esse resultado inviabiliza o uso da *Faster-RCNN* em robôs que necessitam de identificação de objetos em tempo real e que não possuem GPUs.

Tabela I: Comparativo da performace dos três modelos treinados em termos de *frames por segundo*

Modelo	**FPS**
MobileNetV2Quantized	11.03
MobileNetV2	9.64
F-RCNN	0.77

CONCLUSÃO

A *RoboCup* e a área robótica realmente geram desafios que incentivam a comunidade científica a evoluir e consequentemente desenvolver novas técnicas para resolução dos mais diversos problemas, sendo a visão computacional apenas um deles. As redes neurais convolucionais se mostram cada vez mais consolidadas no ramo da visão artificial. Evoluindo tanto no quesito precisão, como pudemos observar com a *Faster R-CNN*, quanto no quesito velocidade, que é o caso das *MobileNets*.

Com o experimento realizado, a *MobileNetV2* foi a que apresentou melhor balanço entre desempenho e precisão. Porém a *MobileNetV2Quantized* é aproximadamente 15% mais veloz em termos de FPS, em detrimento de uma baixa perda na precisão, se considerarmos a mAP com 50% de IoU.

O objetivo de experimentar arquiteturas estado da arte em velocidade e precisão, avaliando-as com métricas *benchmark* difundidas na maioria dos estudos de detecção de objetos foi bem sucedido. Estudos futuros, agora podem ser feitos para comparar em mais profundidade as *MobileNets* em termos de performance e também para treinar as redes por mais passos para avaliar até qual patamar de precisão pode-se atingir. Além disso, as redes treinadas neste artigo podem ser incrementadas para detecção de robôs adversários, pontos de interesse no campo, entre outros, utilizando técnicas de transferência de aprendizado.

REFERÊNCIAS

[1] J. Canas, D. Puig, E. Perdices and T. Gonzalez. “Visual Goal Detection for the RoboCup Standard Platform League.”, 2009.

[2] N. Dalal, B. Triggs. ”Histograms of Oriented Gradients for Human Detection”. International Conference on Computer Vision Pattern Recognition (CVPR ’05), Jun 2005, San Diego, United States. pp.886–893.

[3] P. Viola, M. Jones, ”Rapid object detection using a boosted cascade of simple features”, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei. ”ImageNet Large Scale Visual Recognition Challenge”. IJCV, 2015.

[5] K. Simonyan, A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” CoRR abs/1409.1556, 2015.

[6] A. Krizhevsky, I. Sutskever, G. E. Hinton. 2012. ”ImageNet classification with deep convolutional neural networks”. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS’12), F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 1. Curran Associates Inc., USA, 1097-1105.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich. ”Going deeper with convolutions”. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, pages 1–9. IEEE Computer Society, 2015.

[8] K. He, X. Zhang, S. Ren, J. Sun. ”Deep residual learning for image recognition”. CoRR, abs/1512.03385, 2015.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen. “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[10] D. Speck, P. Barros, C. Weber, S. Wermter. ”Ball Localization for Robocup Soccer Using Convolutional Neural Networks”, RoboCup, 2016.

[11] J. Menashe, J. Kelle, K. Genter, J. Hanna, E. Liebman, S. Narvekar, R. Zhang, P. Stone. ”Fast and Precise Black and White Ball Detection for RoboCup Soccer”, 2018. Lecture Notes in Computer Science, 45–58.

[12] N. Cruz, K. Lobos-Tsunekawa, J. Ruiz-del-Solar. “Using Convolutional Neural Networks in Robots with Limited Computational Resources: Detecting NAO Robots While Playing Soccer.” RoboCup 2017: Robot World Cup XXI. Lecture Notes in Computer Science, pp. 19–30.

[13] M. Buric, M. Pobar, M. Ivašić-Kos, Marina. ”Adapting YOLO Network for Ball and Player Detection”, 2019. 845-851.

[14] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. ”You Only Look Once: Unified, Real-Time Object Detection”, 2016. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.

[15] R. Girshick, J. Donahue, T. Darrell, J. Malik. ”Rich feature hierarchies for accurate object detection and semantic segmentation”. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pages 580–587. IEEE, 2014.

[16] S. Ren, K. He, R. Girshick and J. Sun, ”Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,”in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[17] A. Géron. ”Hands–On Machine Learning with Scikit–Learn and TensorFlow”, 2019, O’Reilly Media.

[18] C. Murch, S. Chalup. ”Combining edge detection and colour segmentation in the four-legged league”. In: Australasian Conference on Robotics and Automation (ACRA 2004) (2004).

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. A. less. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.” CoRR (2017).

[20] J. Wu, C. Leng, Y. Wang, Q. Hu, J. Cheng. “Quantized Convolutional Neural Networks for Mobile Devices.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016).

[21] X. Chen, H. Mulam. “An Implementation of Faster RCNN with Study for Region Sampling.” CoRR abs/1702.02138 (2017).

[22] D. Cook, K. Feuz, N. Krishnan. ”Transfer learning for activity recognition: a survey”, 2013. Knowledge and Information Systems, 36, 537-556.

[23] L. Abreu. ”Object Detection Demo”, 2019, GitHub repository, https://github.com/labreu/object_detection_demo

[24] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng. ”TensorFlow: Large-scale machine learning on heterogeneous systems”, 2015. Software available from tensorflow.org.

[25] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, J. Murphy. ”Speed/accuracy trade-offs for modern convolutional object detectors.”, 2017, CVPR.

[26] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. Zitnick. “Microsoft COCO: Common Objects in Context.”, 2014. Lecture Notes in Computer Science: 740–755.

[27] M. Everingham, S. Eslami, L. Gool, C. Williams, J. Winn, A. Zisserman. ”The Pascal Visual Object Classes (VOC) challenge - a Retrospective”, 2014. IJCV.