ESTUDO, IMPLEMENTO E COMPARAÇÃO DA YOLOV7 E VITDET NO FUTEBOL DE ROBÔS HUMANOIDES

Luana Watanabe Tonieti[1], Reinaldo Augusto da Costa Bianchi[2]
[1] Engenharia de Robôs, Centro Universitário FEI
[2] Engenharia Elétrica, Centro Universitário FEI
luanawt43@gmail.com, rbianchi@fei.edu.br

Resumo: O presente artigo tem como objetivo o estudo de dois tipos diferentes de redes para detecção de objetos (redes neurais convolucionais: YOLOv7; *vision transformer*: ViTDet), a implementação de ambas no código da equipe RoboFEI e a comparação para decidir qual seria a melhor em um jogo de futebol de robôs humanoides. No geral, ViTDet apresentou resultados inferiores, sendo mais devagar que os modelos da YOLOv7 e classificando robôs como bola. O modelo da YOLOv7-*tiny* foi considerado o ideal por sua rapidez.

1. Introdução

O presente trabalho de iniciação científica tem como objetivo estudar e comparar o desempenho de dois tipos diferentes de redes de detecção de objetos: uma que tem o seu funcionamento baseado em redes neurais convolucionais (CNNs), a YOLOv7 [1], e outra que tem seu funcionamento baseado nos v*ision transformers (ViTs)*, a ViTDet [2], ambas sendo aplicadas no futebol de robôs.

Com um funcionamento bem diferente quando comparado com as CNNs, as redes baseadas nos ViTs vêm sendo alvos de diversas pesquisas explorando o potencial que esta rede pode atingir. Tendo isso em vista, com este estudo será possível ter o conhecimento se, na prática, tais redes são capazes de superar as CNNs no âmbito do futebol de robôs humanoides. Para isto, tanto as YOLOv7 quanto a ViTDet serão treinadas com o mesmo *dataset* e serão testadas considerando as mesmas métricas. Por fim, também serão implementadas no atual código da equipe de futebol de robôs humanoides RoboFEI, que atualmente usa o framework ROS2 [3], para serem testadas simulando um jogo real, visto que o objetivo é a detecção de objetos em tempo real.

2. Metodologia

Primeiramente, um *dataset* de imagens de bolas e robôs foi criado pela equipe da RoboFEI, ao qual também foram adicionadas imagens do *dataset* chamado TORSO-21 [4], criado pela equipe de futebol de robôs humanoides *Hamburg Bit-Bots*. Depois, os modelos YOLOv7-*tiny*, YOLOv7 e YOLOv7-X foram escolhidos e treinados com 500 épocas, uma entrada com resolução de 640px e um *batch size* de 16, com a utilização da técnica de aprendizado por transferência, e contou com 46905 imagens para o treinamento, 4461 para a validação e 2255 para os testes. Esta rede foi então testada utilizando as métricas da precisão, a revocação, a acurácia, o F-score, a interseção sobre união (IoU), a precisão média (AP), média da AP (mAP) e o tempo de inferência. Por fim, esta rede foi implementada no código da equipe para funcionar utilizando o ROS2 em um Intel® NUC i5-10210.

Os modelos disponíveis da ViTDet tinham como *head* as redes *Mask*-RCNN e *Cascade Mask* R-CNN. A divisão do *dataset* foi feita considerando 53859 imagens para o treinamento e 4484 imagens para o teste. O treinamento do modelo base da ViTDet com a *Mask* R-CNN como *head* foi feito com 5000 épocas, uma imagem de entrada com resolução de 1024px, uma taxa de aprendizado de 0,0005, um *window size* de 14x14px, um total de 12 *attention heads* e contou com a técnica de aprendizado por transferência. Para o treinamento com a *Cascade Mask* R-CNN como *head* da ViTDet, a única alteração feita foi diminuir a taxa de aprendizado para 0,00005 para garantir a convergência do modelo. Estes modelos também foram testados e implementados com ROS2.

Para medir os tempos de inferência de cada rede, foi utilizada a imagem apresentada na Figura 1, onde as redes deveriam ser capazes de detectar os três robôs e quatro bolas presentes na imagem. Neste caso, todas as inferências foram feitas considerando um valor de limiar para a confiança de 0.45. Um último teste foi feito para medir os tempos de inferência de todas as redes com o uso de uma GPU (NVIDIA *GeForce* GTX 1060 *Mobile*).

Figura 1 – Imagem utilizada para a medição dos tempos de inferência das redes.

3. Resultados e Discussões

A Tabela I apresenta as métricas, com exceção do tempo de inferência, obtidas a partir do teste feito com os modelos YOLOv7-tiny, YOLOv7 e YOLOv7x. No geral, os três modelos apresentaram métricas satisfatórias, conseguindo manter uma AP alta até mesmo para a classificação de robôs.

Os resultados do teste feito com a rede ViTDet podem ser vistos na Tabela II. Comparando com as métricas da rede anterior, é possível perceber que esta rede apresentou resultados inferiores, principalmente em relação a classificação de robôs.

Tabela I – Métricas obtidas pelos três diferentes modelos da YOLOv7.

	YOLOv7-*tiny*	**YOLOv7**	**YOLOv7-X**
Precisão	0.7951	0.7821	0.7693
Revocação	0.9975	0.9998	1.0
Acurácia	0.8222	0.8061	0.7879
F-*score*	0.8772	0.8690	0.8600
IoU	0.6531	0.8938	0.6611
AP Bola	0.8783	0.8935	0.8931
AP Robôs	0.7017	0.7169	0.7204
mAP	0.7900	0.8052	0.8067

Tabela II – Métricas obtidas pela ViTDet.

	ViTDet *Mask* R-CNN	**ViTDet *Cascade***
Precisão	0.5935	0.7615
Revocação	0.5262	0.7564
Acurácia	0.9155	0.9570
F-*score*	0.5315	0.7587
IoU	0.7347	0.7289
AP Bola	0.9040	0.9209
AP Robôs	0.0265	0.3568
mAP	0.4652	0.6389

Para demonstrar o funcionamento destas redes em tempo real, foi gravado um vídeo presente no link [3]. Neste vídeo, o código captura o frame da *webcam* do robô, passa a imagem pela rede para que ela possa fazer uma predição e publica em um tópico a posição do robô ou da bola no frame. O vídeo foi gravado com a webcam que o robô da equipe utiliza atualmente nos jogos usando o NUC dele. Nota-se que os três modelos da YOLOv7 foram capazes de detectar ambos os robôs e a bola precisamente. Ademais, o impacto da quantidade de camadas fica visível ao comparar os trechos dos modelos da YOLOv7-tiny e da YOLOv7x, sendo que nesta última, apesar de ter feitos ótimas predições, demorava bastante, dando a impressão que o vídeo travava. No caso da ViTDet rodando na CPU do NUC, é possível ver que mesmo acelerando o vídeo x2, a rede é extremamente lenta. No entrando, ao rodar na GPU, como visto no final do vídeo, a rede passa a ter um tempo de inferência aceitável, parecidos com os da YOLOv7 e YOLOv7x. No caso da ViTDet com a *Mask* R-CNN como *head*, nota-se que a rede classificou os robôs como bola na maior parte do tempo, o que faz sentido ao verificar as métricas não muito boas apresentadas pela mesma na Tabela II.

Por fim, a Tabela III traz um resumo dos resultados relacionados ao tempo de inferências dessas redes quando rodadas tanto em CPU quanto em GPU e a quantidade de bolas e robôs detectados na Figura 1. Em todos os casos, a GPU trouxe uma melhora significativa para seus tempos de inferência, mas vale destacar a melhora para a ViTDet, visto que para ambas as *heads* houve uma diminuição de mais de 12 segundos.

Tabela III – Resultados das redes em relação ao tempo de inferência.

	Tempo de inferência (CPU)	**Tempo de inferência (GPU)**	**Bolas detectadas**	**Robôs detectados**
YOLOv7 *-tiny*	0.207	0.073	3	2
YOLOv7	0.550	0.237	4	2
YOLOv7 -X	0.845	0.325	4	3
ViTDet *Mask*	13.863	1.347	5	1
ViTDet *Cascade*	15.170	1.743	4	1

4. Conclusões

Apesar dos resultados aceitáveis da ViTDet em relação às métricas, por ser uma rede demasiadamente lenta quando rodada na CPU, a utilização desta rede durante os jogos acaba sendo totalmente inviável, como foi possível ver no vídeo apresentado na seção 3. O menor modelo desta rede acabou sendo ainda mais devagar que a YOLOv7-X, o maior modelo da YOLOv7 analisado. Em uma partida de futebol de robôs humanoides, em que a bola e os robôs estão sempre em movimento, um tempo de inferência muito alto pode ser extremamente prejudicial à equipe, visto que as decisões tomadas pelo robô seriam afetadas por conta do atraso da rede em informar onde a bola e os robôs se encontram.

Com isso, nota-se que o modelo YOLOv7-tiny pode ser considerado o ideal para a equipe, pois além de ser extremamente rápido, também é capaz de, na maior parte do tempo, detectar corretamente a bola e os robôs presentes na imagem.

5. Referências

[1] WANG, Chien-Yao; BOCHKOVSKIY, Alexey; LIAO, Hong-Yuan Mark. **YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors**. [S.l.]: arXiv, 2022. DOI: 10.48550/ARXIV.2207.02696. Disponível em: https://arxiv.org/abs/2207.02696.

[2] LI, Yanghao et al. **Exploring Plain Vision Transformer Backbones for Object Detection**. [S.l.]: arXiv, 2022. DOI: 10.48550/ARXIV.2203.16527. Disponível em: https://arxiv.org/abs/2203.16527.

[3] MACENSKI, Steven et al. Robot Operating System 2: Design, architecture, and uses in the wild. **Science Robotics**, v. 7, n. 66, eabm6074, 2022. DOI: 10.1126/scirobotics.abm6074. Disponível em: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074.

[4] BESTMANN, Marc et al. **TORSO-21 Dataset: Typical Objects in RoboCup Soccer 2021**. [S.l.: s.n.], 2021. Disponível em: https://github.com/bit-bots/TORSO_21_dataset.

[1] Aluna de IC do Centro Universitário FEI. Projeto com vigência de 05/2022 a 04/2023.

[3] https://youtu.be/5J1AxCly_V8