

USO DE CNNs PARA A DETECÇÃO DE PONTOS DE REFERÊNCIA VISUAIS PARA UM ROBÔ HUMANOIDE

Massiel Blandy Ramón¹, Reinaldo Augusto da Costa Bianchi²

¹ Departamento de Engenharia de Elétrica, Centro Universitário FEI

² Departamento de Engenharia Elétrica, Centro Universitário FEI

uniemramon@fei.edu.br, rbianchi@fei.edu.br

Resumo: No projeto desenvolveu-se um algoritmo de detecção de *landmarks* em campos de futebol de robôs humanoides utilizando Redes Neurais Convolucionais (CNNs). Implementado nos robôs da equipe RoboFEI, o sistema identifica as linhas do centro, do pênalti e as traves do gol, calculando a distância até esses pontos por métodos trigonométricos. Os resultados demonstraram alta eficácia tanto no treinamento quanto na detecção e no cálculo de distâncias, com desempenho ligeiramente inferior na detecção das traves do gol.

1. Introdução

A evolução dos sistemas robóticos tem desempenhado um papel crucial na melhoria da qualidade de vida e na expansão das fronteiras tecnológicas. Este projeto visa aprimorar a autonomia de robôs humanoides em competições de futebol, como a RoboCup, por meio da utilização de CNNs para a detecção de pontos de referência visuais, como as linhas do campo e as traves do gol. A implementação desse sistema nos robôs da equipe RoboFEI busca melhorar a localização e a tomada de decisões durante as partidas, contribuindo para um desempenho mais eficiente dos robôs.

2. Revisão Bibliográfica

Visão computacional é um campo da inteligência artificial que lida com o desenvolvimento de algoritmos e técnicas para permitir que computadores interpretem e compreendam imagens e vídeos. [1]

Uns dos principais componentes da visão computacional são as Redes Neurais Convolucionais (CNNs), arquiteturas de aprendizado profundo projetadas especificamente para processar dados visuais. Elas são especialmente eficazes em reconhecer padrões, como formas e texturas, em imagens, tornando-se fundamentais para tarefas como detecção de objetos, reconhecimento facial e classificação de imagens. As CNNs geralmente incluem três tipos principais de camadas: convolução, *pooling* e ReLU. [2]

3. Metodologia

3.1. Criação do dataset e treinamento da rede

Foi criado um *dataset* de imagens do campo de futebol de robôs humanoides, capturadas durante competições como a RoboCup e a LARC, utilizando a câmera Logitech C920. As imagens foram anotadas no Roboflow, focando em três classes principais: linhas do centro, linhas de pênalti e traves do gol.

A rede escolhida para o projeto foi a YOLOv7. O treinamento foi realizado utilizando um *dataset* dividido

em 16.572 imagens para treinamento e 611 para validação. Com 500 épocas de treinamento e técnicas de *data augmentation*, o modelo foi ajustado para reconhecer os pontos de referência, mesmo em condições variadas. O código de treinamento encontra-se no site do GitHubⁱ especificado no final do artigo.

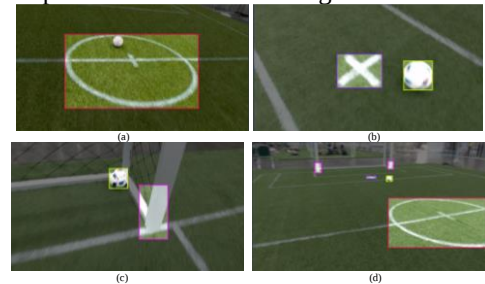


Figura 1 – Anotações realizadas no Roboflow das três classes: (a) *center*, (b) *penalty* e (c) *goalpost*.

3.2. Cálculo da distância e implementação no robô

O cálculo foi realizado utilizando trigonometria aplicada em triângulos, uma abordagem simples e econômica que evita o uso de sensores adicionais e é adequada aos dados disponíveis: a altura do robô até o pescoço quando está parado em sua posição inicial (h_r), a altura do motor do pescoço até o centro da câmera (h_c), e o ângulo formado pelo motor do pescoço e uma linha horizontal paralela ao chão (α).

O método empregado requereu a centralização do ponto de referência no campo de visão vertical da câmera, alinhando a linha de visão do robô a um ângulo de 90 graus com a linha que conecta o centro da câmera ao motor do pescoço. Os motores utilizados no pescoço dos robôs Altair e Atom são Dynamixel XM430. Com o software Dynamixel foi possível mapear as posições dos motores, onde 1024 corresponde a 0 graus e 2048 a 90 graus. Com base nisso, desenvolveu-se a seguinte fórmula para converter a posição do motor em ângulo:

$$\text{ângulo} = \frac{(\text{posição} - 1024) \cdot 90}{1024} \quad (1)$$

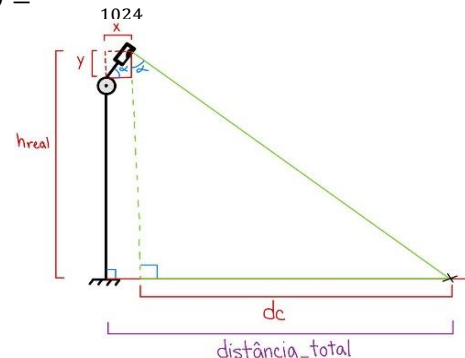


Figura 2 – Esboço simplificado da análise feita para cálculo de distância entre robô e *landmark*.

Com h_c e α conhecidos, calculam-se as componentes x e y , resultantes do deslocamento horizontal e vertical da cabeça do robô quando o motor do pescoço se move:

$$x = \cos \alpha \cdot h_c \quad (2)$$

$$y = \sin \alpha \cdot h_c \quad (3)$$

A altura real do robô quando o motor do pescoço gira é representada por:

$$h_{real} = h_r + y \quad (4)$$

Com isso, calcula-se d_c e, depois, a distância total:

$$d_c = \tan \alpha \cdot h_{real} \quad (5)$$

$$distância_total = d_c + x \quad (6)$$

A integração do projeto no robô efetuou-se através do código disponível no GitHubⁱⁱ da autora. Recomenda-se a leitura do README do repositório.

4. Resultados e discussão

O treinamento da rede YOLOv7 foi bem-sucedido, apresentando altos índices de confiança na detecção de *landmarks* tratados. A matriz de confusão revelou uma alta precisão, com 97% para o centro, 92% para as traves do gol e 96% para a marca de pênalti. No entanto, houve uma confusão leve na detecção das traves do gol, com algumas previsões incorretamente classificadas como fundo.

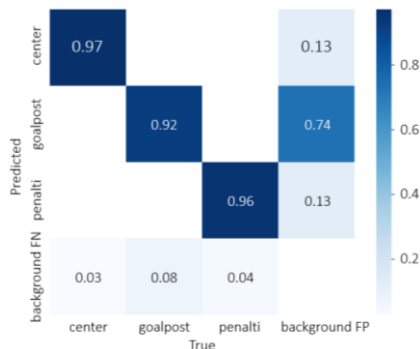


Figura 3 – Matriz de confusão para o desempenho da rede YOLOv7 treinada.

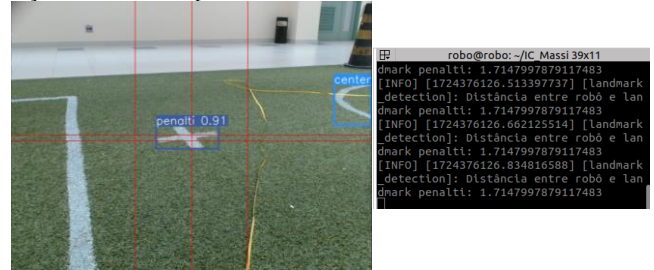
A rede mostrou-se eficaz na generalização, obtendo bons resultados em testes com imagens inéditas, embora com uma confiança moderada na detecção das traves. Isso ocorreu devido ao uso de um gol diferente nos testes, em comparação ao utilizado durante o treinamento, impactando a familiaridade do modelo com o ambiente. Para melhorar isso, seria necessário aumentar o número de imagens de traves e incluir diferentes tipos de gols.



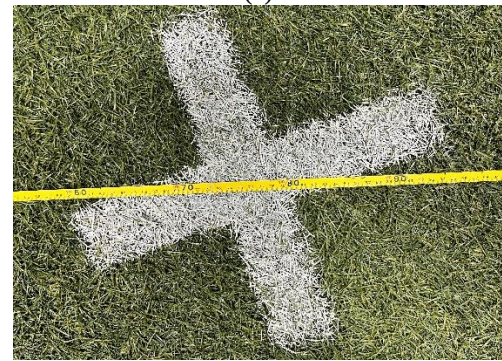
Figura 4 – Exemplo de resultados obtidos no campo da RoboFEI.

Para o cálculo da distância entre robô e *landmark* os resultados foram satisfatórios, mostrando uma

aproximação significativa entre os valores calculados e os reais. O cálculo foi mais preciso para o pênalti, enquanto a trave apresentou uma ligeira maior margem de erro. Embora alta precisão não fosse essencial para o projeto, os resultados superaram as expectativas. A aplicação deste método revelou-se prática e eficaz, embora alternativas como sensores de profundidade ou visão estéreo possam oferecer maior precisão em outros contextos. No link do Youtubeⁱⁱⁱ fornecido no final deste documento apresenta-se um vídeo demonstrativo das medições realizadas para os três *landmarks*.



(a)



(b)

Figura 5 – Exemplo de medição da distância entre robô e pênalti (a) valor calculado (b) valor real.

5. Conclusões

O projeto obteve resultados promissores ao implementar um sistema de detecção de *landmarks* em robôs humanoides utilizando CNNs (YOLOv7). O modelo apresentou alta precisão na detecção dos pontos de referência, com taxas de confiança que superaram o 85% na maioria dos testes. O método trigonométrico empregado para calcular a distância proporcionou resultados que superaram as expectativas. Esses avanços demonstram a viabilidade de soluções baseadas em visão computacional para aplicações robóticas complexas.

6. Referências

- [1] SZELISKI, Richard. **Computer Vision: Algorithms and Applications**. 2.ed. Springer, 2021.
- [2] **Convolutional Neural Network**. Towards Data Science, 24 fev. 2019. Disponível em: <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529>. Acesso em 5 mar. 2023.

¹ Aluno de IC do Centro Universitário FEI. Projeto com vigência de 06/2023 a 05/2024.

ⁱⁱⁱ <https://youtu.be/ggnsLBdY20?si=FCKUgbrQhvCTuH8>

ⁱ <https://github.com/WongKinYiu/yolov7>

ⁱⁱ https://github.com/massiblandy/Scientific_Research_Project/blob/main/src/visi_on_yolov7/vision_yolov7/detect.py