Desenvolvimento de um sistema de visão computacional aplicado ao futebol de robôs

Vitor da Silva Dias[1], João Carlos Nunes Bittencourt[2]

[1]Bacharelado em ciências exatas e tecnológicas – Universidade Federal do Recôncavo da Bahia – Campus Cruz das Almas – Cruz das Almas – BA – Brazil

[2]Engenharia de Computação – Universidade Estadual de Feira de Santana – Campus Feira de Santana – Feira de Santana – BA – Brazil

Vitordias@aluno.ufrb.edu.br, Joaocarlos@ufrb.edu.br

Abstract. *In the IEEE Very Small Size category of robot soccer, one of the challenges is to determine the position coordinates of both the robots and the ball throughout a match. For this, in general, it is necessary to use computer vision techniques. In a match, robots adopt custom markers from standard colors. The solution proposed by this work uses two models of Convolutional Neural Networks based on the real-time object detection system You only look once (YOLO), which works according to time markers, to efficiently track the robots and the ball. The system operates at a rate of 2 FPS in player detection and 30 FPS in time and ball identification, returning the position of each detected object.*

Resumo. *Na categoria IEEE Very Small Size de futebol de robôs, um dos desafios é determinar as coordenadas da posição tanto dos robôs quanto da bola ao longo de uma partida. Para isso, em geral, se faz necessário utilizar técnicas de visão computacional. Em uma partida, os robôs adotam marcadores customizados a partir de cores padrão. A solução proposta por esse trabalho utiliza dois modelos de Redes Neurais Convolucionais baseados no sistema detecção de objetos em tempo real You Only Look Once (YOLO), que trabalham de acordo com os marcadores do time, para rastrear, de forma eficiente, os robôs e a bola. O sistema opera a uma taxa de 2 FPS na detecção dos jogadores e 30 FPS na identificação dos times e da bola retornando a posição de cada objeto detectado.*

1. Introdução

Na história do futebol de robôs, algoritmos tradicionais de segmentação de cores e detecção de bordas foram amplamente utilizados para atingirem resultados cada vez mais satisfatórios nas competições. Entretanto, mudanças recentes nas regras, como a permissão de no máximo metade da bola ser qualquer combinação de cores e a alternância entre luz artificial e luz natural, fizeram com que os algoritmos tradicionais fossem afetados significativamente, uma vez que a distribuição de cores da bola pode variar com base no ângulo e na iluminação do ambiente[Militão e Colombini, 2017].

Nos últimos anos, os avanços em aprendizagem profunda levaram a uma mudança de paradigma no campo de visão computacional, permitindo a realização da detecção de inúmeras classes de objetos em tempo real cada vez mais precisas e confiáveis, sendo este um requisito fundamental para que o desafio proposto pela RoboCup seja alcançado. Com isso, o estudo da aplicação de Redes Neurais Convolucionais no universo do futebol de robôs se torna interessante, dada a sua comprovada eficiência em aplicações relacionadas[Militão e Colombini, 2017]. Nesse contexto, o presente artigo apresenta um sistema de visão computacional aplicado ao futebol de robôs, capaz de rastrear os robôs jogadores e a bola em tempo real, em um time da categoria Very Small Size, auxiliando assim na navegação dos robôs.

2. Fundamentação Teórica

2.1 Futebol de Robôs.

O futebol de robôs é um desafio da robótica móvel reconhecido mundialmente e amplamente discutido nos fóruns científicos [Tadokoro, 2000; Akin, 2013]. A primeira iniciativa no sentido do futebol de robˆos foi proposta a d´ecada de 1990 como forma de incentivar pesquisas e experimentos no ambiente acadêmico na temática de robótica autônoma multiagente [Hoopes, 2003]. Para Impulsionar essa meta, pesquisadores propuseram, como objetivo principal do futebol de robôs, o desenvolvimento de um time de robôs humanóides autônomos capaz de vencer a seleção˜ao campeã da FIFA, até meados do século XXI [Kitano, 1997; Kitano, 1998; Akin, 2013; Holz, 2019].

O futebol de robôs foi proposto por pesquisadores com o objetivo de criar um novo desafio científico de longo prazo, a partir do desenvolvimento de times de robôs envolvendo integração de técnicas de Inteligência Artificial [Kitano e Sanderson 1997]. Segundo [Kraetzchmar, 1998], “Dispositivos mecatrônicos, hardware especializado para o controle de sensores e atuadores, teoria de controle, interpretação e fusão sensorial, redes neurais, computação evolutiva, visão, e sistemas multi-agentes são exemplos de campos envolvidos nesse desafio”. Deste modo, observa-se que os desafios presentes no futebol de robôs contribuem de forma significativa para diversos desafios tecnológicos da automação e da robótica móvel.

A IEEE Very Small Size (VSS) é uma das categorias do futebol de robôs, na qual cada time é formado por três robôs (assumindo as posições de goleiro, atacante e defensor), que devem ser autônomos, havendo comunicação apenas com um computador destinado ao controle e processamento dos dados coletados da partida. A Figura 1 ilustra a estrutura física associada a uma partida de futebol de robôs na categoria VSS.

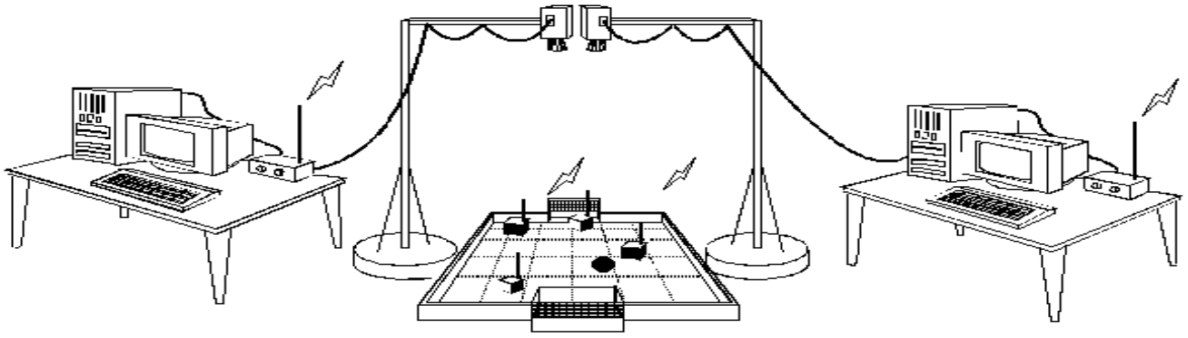


Figura 1: Estrutura da categoria Very Small Size

Fonte: Acervo do google

De acordo com as regras da modalidade, cada time é identificado com uma cor obrigatória que deve estar presente no marcador de identificação de cada robô, podendo cada time assumir as amarelo ou azul. Na parte superior do campo, uma câmera capta a informação visual de todo o campo, incluindo a movimentação dos jogadores e da bola. O computador realiza o processamento e os cálculos de navegação e posicionamento de cada jogador dentro do campo.

2.2 Visão Computacional

Desde as primeiras concepções sobre redes neurais artificiais, datadas de 1943, por Warren McCulloch e Walter Pitts [Fleck, 2016], os estudos em neurocomputação avançando de tal forma a permitirem grandes avanços, que hoje possibilitam a resolução de problemas complexos como a detecção de objetos em tempo real por meio de aprendizagem profunda.

A Visão Computacional tem como principal objetivo fornecer aos computadores uma capacidade de percepção semelhante ou até superior à visão dos seres humanos. De maneira mais precisa, ela visa o desenvolvimento de técnicas que permitam extrair informações e características relevantes do mundo real a partir de uma imagem [Khan, 2018].

O objetivo principal da visão computacional é entender a história por detrás de uma imagem. Para o ser humano, essa tarefa é simples de ser realizada, mas para computadores se apresenta como uma tarefa extremamente complicada [Coldewey e Devin 2016]. Uma das tarefas largamente associadas à Visão Computacional é a detecção de objetos. Esta tarefa visa localizar e separar objetos de interesse contidos na imagem. O estado da arte da detecção de objetos atualmente é o sistema You Only Look Once (YOLO), o qual utiliza técnicas de aprendizagem máquina para localizar objetos [Redmon e Farhadi, 2018].

2.3 You Only Look Once

Uma Rede Neural Artificial (RNA) é uma unidade paralela e distribuída, constituída de unidades de processamento simples, cujo objetivo é armazenar o conhecimento experimental e torná-lo disponível para o uso [Haykin, 1998]. Redes Neurais Convolucionais (CNN) são uma das categorias mais populares de RNA, especialmente para dados multidimensionais, a exemplo de imagens e vídeos. Uma CNN opera de maneira semelhante às RNA. A diferença fundamental reside no fato de que cada unidade em uma camada da CNN é um filtro

bidimensional o qual é convoluído com a entrada dessa camada. Esse processo é essencial para os casos em que o sistema precisa aprender padrões de mídia de entrada multidimensionais. Ao longo das camadas, a rede vai aprendendo e extraindo padrões mais complexos [Khan, 2018].

O modelo de CNN baseado no sistema You Only Look Once (YOLO) é um sistema de detecção de objetos em tempo real de alta eficiência [Redmon e Farhadi 2018]. Enquanto os métodos tradicionais de detecção de objeto precisam executar o classificador centenas ou milhares de vezes, o YOLO se sobressai por ser um método de detecção de passada única, que utiliza uma CNN como um extrator de características. De acordo Redmon e Farhadi (2018), o YOLO aplica uma única rede neural para toda a imagem, dividindo esta em regiões delimitadas por caixas. Para cada caixa é dada uma probabilidade de um objeto estar contido nela. As áreas com maiores probabilidades são consideradas como áreas em que há objetos de interesse. Nos últimos anos YOLO tem sido apresentado como uma técnica de alto desempenho e eficiência para resolução de problemas que têm como requisito alto desempenho, devido aos seus resultados atrelados à baixa latência.

Segundo Redmon e Divvala (2015), no algoritmo YOLO, a imagem de entrada é dividida em uma grade de tamanho *S x S*. Se o centro de um objeto estiver dentro de uma célula dessa grade, tal célula é responsável por detectar o objeto. Cada célula da grade gera um número *B* de caixas delimitadoras e uma pontuação de confiança da caixa. Essa pontuação de confiança diz o quão confiável é o formato daquela caixa para prever o objeto de maneira precisa. Se não há objeto na célula, a pontuação de confiança é zero. Caso contrário, essa pontuação será igual à interseção sobre união entre a caixa gerada e a caixa verdadeira. A Figura 2 mostra de maneira resumida o método de detecção YOLO.

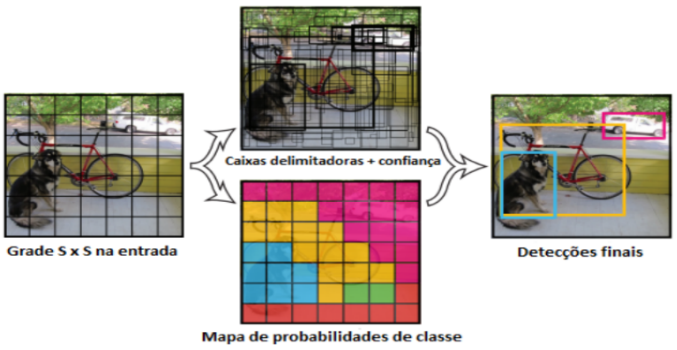


Figura 2: Método de detecção YOLO.
Fonte: Adaptado de Redmond e Divvala (2015)

3. Sistema de Visão Computacional VSS

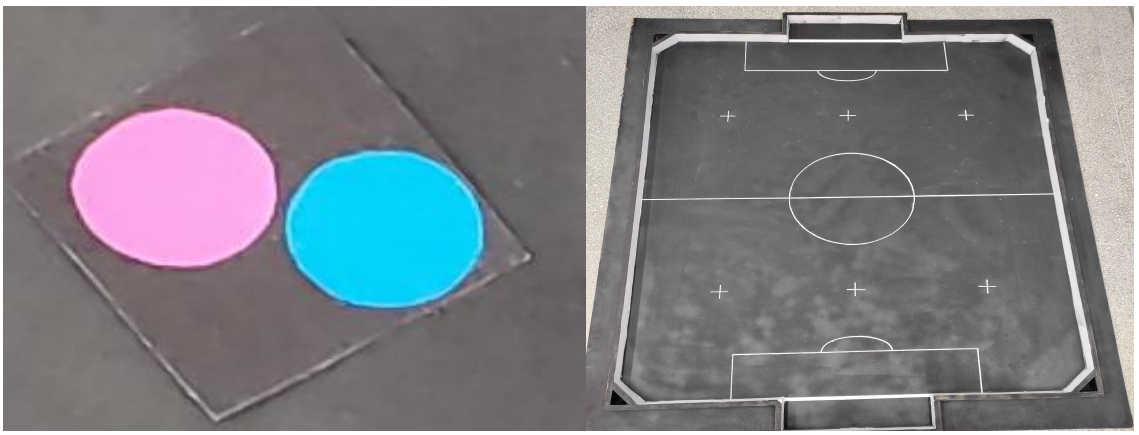
A abordagem metodológica deste trabalho teve como base os requisitos necessários para o desenvolvimento de um sistema de visão computacional para a categoria VSS de futebol de robôs. Nesse sentido, foi considerada a necessidade de proporcionar um sistema capaz de detectar, rastrear e identificar os robôs e a bola dentro do campo. A partir desses requisitos, foi possível estabelecer uma diretriz inicial para o funcionamento do sistema, para então fazer

conjecturas sobre os resultados iniciais. Com base nisso, a revisão de literatura partiu do estudo das técnicas mais usadas para resolver os problemas similares, buscando entender quais as métricas de avaliação e técnicas que resultam em uma melhor contribuição para o desenvolvimento da aplicação final.

O trabalho consistiu em três pilares: (i) a construção do *dataset* de treinamento e validação da rede neural; (ii) o treinamento da Rede Neural Convolucional; e (iii) a validação da inferência e testes do sistema em um ambiente real. O sistema funciona utilizando dois modelos de CNN. A primeira rede detecta a bola e determina a qual time o jogador pertence, de acordo com a cor dos marcadores posicionados sobre os robôs. A partir dessa informação, a segunda CNN é responsável por detectar a posição correspondente a cada jogador da sua equipe, determinando se goleiro, zagueiro ou atacante. Essa abordagem tem como objetivo aumentar a precisão e a confiança na detecção dos jogadores, proporcionando assim uma melhor estimativa de desempenho da equipe nas partidas disputadas.

3.1 Construção do *Dataset*

Um *dataset* é um conjunto de dados, para o contexto do projeto de um RNA. Esses dados são determinados a partir de características do objeto a ser detectado, extraídos através de imagens. O presente trabalho construiu dois *datasets* com imagens capturadas pela equipe de desenvolvimento. Para adquirir essas imagens foi necessário construir um cenário em acordo com as regras da categoria VSS; com o padrão de marcadores para os robôs e o campo onde ocorrem as partidas. A Figura 3a ilustra o protótipo do robô com seu respectivo padrão de detecção. A Figura 3b representa o campo construído pela equipe. Vale ressaltar que as fotos não foram capturadas por uma câmera posicionada de forma totalmente perpendicular ao campo.



(a) **(b)**

Figura 3: Artefatos construídos pela equipe.

A partir das imagens capturadas foram realizadas anotações sobre o posicionamento dos objetos que precisavam ser detectados. Essas anotações foram produzidas no formato *Pascal VOC*, e armazenadas em arquivo XML. Esse procedimento foi conduzido a partir do uso da ferramenta *LabelImg*, largamente empregada na construção de *datasets* para redes

neurais artificiais. Nesse processo, foram desenhadas as caixas delimitadoras envolta dos objetos, como descrito na Figura 4.

Figura 4: Processo de anotação de imagem no LabelImg.

O dataset produzido a partir das imagens com suas respectivas anotações foi dividido em conjuntos utilizados treinamento e na validação. Ambos os *datasets* foram produzidos considerando como parâmetro 80% das imagens para treinamento e 20% para validação. O primeiro *dataset* que diz respeito ao modelo que detecta a bola e os times, é composto por 85 imagens para treinamento e 20 para validação. O segundo *dataset* se refere a identificação do jogador, sendo este composto por 228 imagens para treinamento e 39 para validação.

3.2 Treinamento do Modelo

Os modelos foram treinados utilizando a técnica de transferência de aprendizagem através de modelos pré-treinados, utilizando a arquitetura Darknet, uma estrutura de rede neural de código aberto escrita em C e CUDA, uma vez que essa abordagem costuma apresentar bons resultados no processo de inferência, além de reduzir o tempo de treinamento[Khan, 2018]. O conceito de pré-treinamento é amplamente empregado em aplicações de aprendizagem profunda. Uma prática muito bem-sucedida nesses casos é primeiro treinar a rede neural em um problema relacionado, mas diferente, onde uma grande quantidade de dados de treinamento já esteja disponível. Essa técnica faz com que o algoritmo tenha precisão e rapidez ao mesmo tempo, e é por isso que é amplamente utilizado no treinamento de redes neurais profundas [Ruder, 2017]. Posteriormente, o modelo treinado pode ser ajustado para a nova tarefa, inicializando com pesos pré-definidos num conjunto de dados maior. Esse processo é conhecido como *fine tuning* e é uma maneira simples e eficaz de transferir o aprendizado de uma tarefa para outra [Khan 2018].

Foi necessário treinar três modelos de Rede Neural, para que o sistema fosse capaz de se adequar às necessidades da equipe conforme a partida vai se desenhando:

- **Rede 1**: o time adversário assume o marcador na cor amarela.
- **Rede 2**: o time adversário assume o marcador na cor azul.
- **Rede 3:** determina qual o tipo de jogador do time.

As redes foram treinadas no ambiente Google COLAB, uma ferramenta para pesquisa na área de aprendizado de máquina, capaz de realizar o processo de treinamento num tempo menor devido ao alto poder de processamento disponibilizado pelas instâncias da plataforma. O código foi desenvolvido na linguagem Python, versão 4.1.2.30, e as bibliotecas OpenCV e ImageAI, assim como as bibliotecas de rede neural Keras, na versão 2.4.3 e TensorFlow, na versão 2.4.0. As redes foram treinadas por 82 épocas para a Rede 1, 70 épocas para a Rede 2 e 144 épocas para a Rede 3. As épocas correspondem à quantidade de execuções de treinamentos sobre um modelo de dados, adotando um padrão de mini-lotes de tamanho igual a 4 imagens organização em três classes, definidas de acordo com os objetos detectados.

3.3 Validação da Rede Neural

O processo de validação dos modelos treinados consiste em inferir a rede sobre um conjunto de imagens presentes no *dataset*, juntamente com sua respectiva anotação. Esse processo visa relacionar a detecção do modelo com as anotações da imagem, verificando assim se o objeto detectado corresponde àquilo que deveria ser detectado. Através da validação é possível determinar o valor de confiança, verdadeiros positivos (VP), falsos positivos (FP) que o modelo treinado apresenta, além da taxa de quadros não detectados (QN), falsos positivos (FP) e falsos negativos (FN).

A partir dessas taxas, é possível extrair o *Mean Average Precision* (MAP), uma métrica largamente utilizada na avaliação de modelos de redes nerais, através de duas outras métricas utilizadas para avaliar a performance de modelos de classificação: precisão e *recall*. O cálculo da precisão, expresso pela Equação 1, é realizado a partir da razão do número total de casos em que o modelo classificou corretamente um objeto e todos os casos previstos.

$$\text{precisão} = \frac{VP}{VP + FP} \qquad (1)$$

O *recall*, outra métrica utilizada para avaliação de modelos de classificação. Ele é calculado a partir da razão entre todas as previsões corretas sobre o total de verdadeiros positivos reais. Essa expressão é apresentada na Equação 2.

$$\text{recall} = \frac{VP}{VP + FN} \qquad (2)$$

De posse dos valores de precisão e *recall*, é possível projetar um gráfico chamado de curva PR(Precisão Recall), como o apresentado na Figura 5, em que o eixo y é a precisão e o eixo x é o recall.O cálculo da área sob a curva PR é tido como o MAP.

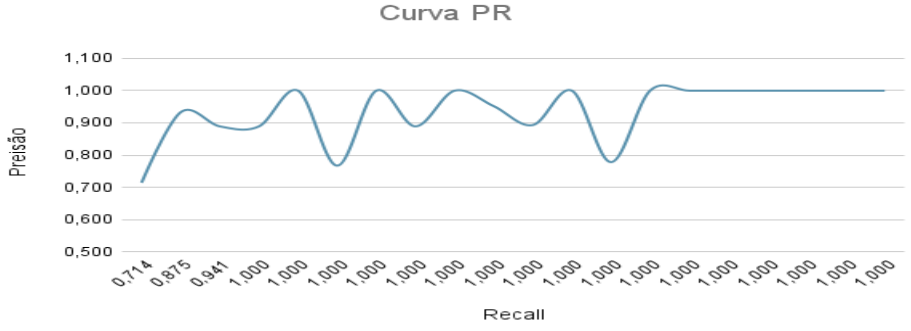


Figura 5: Gráfico da curva PR.

Finalmente, a MAP é obtida realizando a média da precisão calculada para todas as classes de objetos. O algoritmo que realiza o processo de validação retorna a MAP calculada ao fim do processo de validação, que vai de 0 como valor mínimo e 1 como máximo. Esses valores permitem escolher o melhor modelo de acordo com a tarefa de detecção.

Os testes finais foram realizados no ambiente projetado pela equipe [retirado em razão do *blind review*]. Além disso, foram realizados testes com um vídeo produzido durante a realização de um jogo em uma competição oficial. A Seção 4 sintetiza os resultados obtidos a partir dos experimentos realizados nessa etapa.

4. Resultados e Discussões

Para o desenvolvimento do presente trabalho, foram selecionados, modelos pré-treinados para as Redes 1, 2 e 3, para as quais os resultados do treinamento são apresentados na Tabela 1.

Tabela 1: modelos treinados.

	MAP	Épocas
Rede 1	0.7527	80
Rede 2	0.7493	30
Rede 3	0.6479	111

Com propósito de depuração, a cada quadro detectado, foi utilizado um método no qual os modelos exibem na imagem original, a localização, probabilidade e nome do objeto detectado, contribuindo assim para uma melhor visualização dos resultados obtidos pelo sistema. Um exemplo de retorno da Rede 1 é apresentado na Figura 6.

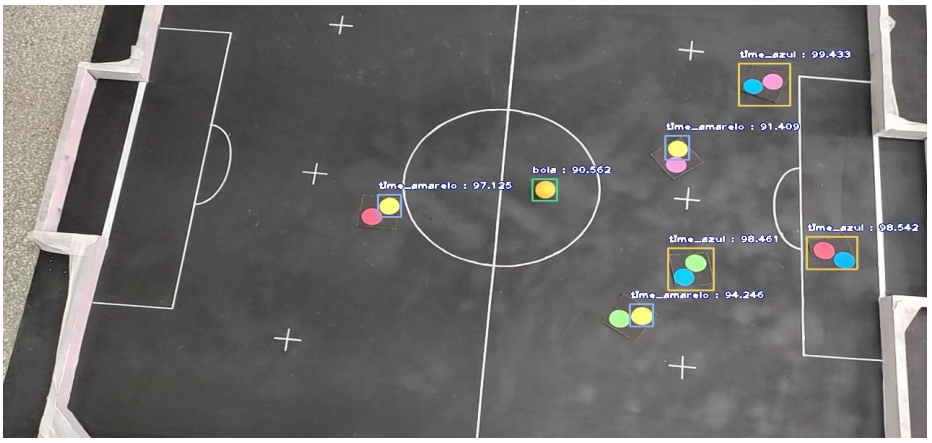


Figura 6: Retorno de forma ilustrativa da rede 1.

Para avaliar os resultados, o sistema foi validado utilizando 5 vídeos. Para cada vídeo, foram retirados 10 quadros de forma aleatória, a fim de obter as métricas de verdadeiro positivo, falso positivo, quadros não detectados e a taxa de probabilidade em cada quadro. Uma síntese desses dados é apresentada na Tabela 2.

Tabela 2: Tabela onde consta as taxas das métricas do sistema.

	Rede 1 (%)	Rede 2 (%)	Rede 3 (%)
VP	91,85	92,55	77,78
FP	30,29	2,89	7,40
FN	2,00	2,62	2,21
QN	2,86	1,94	14,60

A Tabela 3, por sua vez, sistematiza a taxa de acurácia e probabilidade dos robôs e da bola. Observa-se aqui que o sistema alcançou taxas de acurácia acima de 80% para a detecção da bola com probabilidade de 90%. Ao mesmo tempo, a detecção dos robôs chegaram a taxas de 71% de acurácia, com probabilidade de 95%. A menor taxa de acurácia na detecção do robô se justifica pelo fato de serem múltiplos objetos se movendo de forma aleatória e com marcadores diferentes sobre o campo. Esse resultado se deu graças a utilização dos dois modelos de RNC em conjunto, operando a uma taxa de 2 FPS na detecção dos jogadores e 30 FPS na detecção dos times e da bola.

Tabela 3: Taxas de probabilidade e acurácia.

	Robô (%)	Bola (%)	Sistema (%)
Taxa de acurácia	71,43	80,95	76,34
Probabilidade	95,71	90,31	94,95

Ao simular uma partida em tempo real, o sistema foi capaz de operar nas situações em que a equipe adversária é posta com os marcadores amarelos ou azuis. Os testes realizados mostram ainda que o sistema é capaz de rastrear o time adversário independente do formato de seus marcadores, demonstrando assim a alta versatilidade da topologia proposta para o sistema. Finalmente, o sistema se mostrou capaz de mapear a posição de cada objeto detectado na imagem. Essa informação pode ser extraída diretamente da posição do centro da região na qual o objeto foi detectado e pode ser utilizada como referência para o sistema de navegação dos robôs.

5. Conclusão

O presente trabalho apresentou um sistema eficiente e com uma elevada taxa de acurácia para o rastreio da bola e dos robôs, com *datasets* produzidos pela equipe do projeto.Os resultados obtidos a partir da presente investigação demonstram a eficácia da topologia proposta para o sistema de visão computacional para a categoria VSS baseado no sistema YOLO. Esse sistema pode ser utilizado ao longo de uma uma partida de futebol de robôs para a categoria VSS, retornando a posição dos objetos em tempo real. Comprindo assim os objetivos de avaliar a aplicação de técnicas de inteligência artificial ao desenvolvimento do sistema de visão computacional para a categoria IEEE Very Small Size e desenvolver um sistema de

visão computacional eficiente, com baixa latência e alta precisão, auxiliando na navegação dos robôs.

O sistema opera a uma taxa de 2 FPS na detecção dos jogadores e 30 FPS na detecção dos times e da bola, com taxas de acurácia acima de 80% para a detecção da bola com probabilidade de 90%. Ao mesmo tempo, a detecção dos robôs chegaram a taxas de 71% de acurácia, com probabilidade de 95%, demonstrando uma considerável taxa de confiança do sistema. Trabalhos futuros devem buscar métodos para aumentar a taxa de quadros por segundo do sistema. Outra perspectiva de trabalho futuro é determinar o posicionamento do jogador a partir dos dados já fornecidos pelo sistema para auxiliar na navegação dos robôs.

Referências

Braga, A. d. P., Ludermir, T. B., and Carvalho, A. C. P. d. L. F. (2000). Redes neurais artificiais: teoria e aplicações. LTC.

HAN, S. A Guide to Convolutional Neural Networks for Computer Vision. Crawley: Morgan Claypool, 2018.

FLECK, L. Redes Neurais Artificiais: Princípios Básicos. Revista Eletrônica Científica Inovação e Tecnologia - Universidade Tecnológica Federal do Paraná, 2016.

REDMON J.; DIVVALA, S. You Only Look Once: Unified, Real-Time ObjectDetection.arXiv:1506.02640, 2015.

RUDER, S. An overview of gradient descent optimization algorithms. arXiv:1609.04747,2017.

REDMON J.; FARHADI, A. YOLO9000: Better, Faster, Stronger. arXiv:1612.08242, 2016.

KHAN, S. A Guide to Convolutional Neural Networks for Computer Vision. Crawley: Morgan Claypool, 2018.

MILITÃO G.; COLOMBINI, E. RoboCup Soccer Ball Depth Detection using Convolutional Neural Networks. Universidade Estadual de Campinas - Instituto de Computação, 2017.

OLIVEIRA, M.; BOWEN, B.; MCKENNA, R.; CHANG, Y.-S. Fast digital image inpaintingIn: Proceedings of the International Conference on Visualization, Imaging and Image Processing.

Hoopes, D., Davis, T., Norman, K., and Helps, R. (2003). An autonomousmobile robot development platform for teaching a graduate levelmechatronics course. In33rd Annual Frontiers in Education, 2003.FIE 2003., volume 2, F4E–17. IEEE.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E., and Matsu-bara, H. (1997). Robocup: A challenge problem for ai.AI Magazine,18(1), 73–85.

Kitano, H., Asada, M., Noda, I., and Matsubara, H. (1998). Robocup:robot world cup.IEEE Robotics Automation Magazine, 5(3), 30–36.

Tadokoro, S., Kitano, H., Takahashi, T., Noda, I., Matsubara, H., Shin-joh, A., Koto, T., Takeuchi, I., Takahashi, H., Matsuno, F., et al.(2000). The robocup-rescue project: A robotic approach to the di-saster mitigation problem. InProceedings 2000 ICRA. MillenniumConference. IEEE International Conference on Robotics and Auto-mation., volume 4, 4089–4094. IEEE.