

Trabalho final de Inteligência Artificial

Pedro Miguel Pinto Botelho, Ricardo Augusto Silva Bonfim,
Rômulo José Pereira Da Costa Junior,
Thiago Reis Santana

¹Instituto de Computação – Universidade Federal do Amazonas(UFAM)
Av. Gen. Rodrigo Octávio,6200, Coroado I,Setor Norte do Campus–69080–900

{pedro.botelho, ricardo.bonfim, romulo.junior, thiago.santana}@icomp.ufam.edu.br

Abstract. *The semantic interpretation of images is an important task that is applicable in different areas and has various purposes depending on the context. However, achieving this is a complex challenge in some situations, since in some cases it is not just a question of recognizing individual objects, but also of understanding the actions taking place and the spatial relationships between them. In this context, the Neurosymbolic AI approach has become a promising solution. In this work, we will explore the use of Neurosymbolic AI with an approach based on Logical Tensor Networks (LTN). The LTN will be used to identify relationships between bounding boxes in images.*

Resumo. *A interpretação semântica de imagens é uma tarefa importante que é aplicável em diversas áreas e possui diversas finalidades dependendo do contexto. Porém, a sua realização representa um desafio complexo em algumas situações, uma vez que, em alguns casos, não se trata apenas de reconhecer objetos individuais, mas também de compreender as ações em curso e as relações espaciais entre eles. Nesse contexto, a abordagem da IA Neurosímbolica surge como uma solução promissora. Nesse trabalho, iremos explorar o uso de IA Neurosímbolica com uma abordagem baseada em Logical Tensor Networks (LTN). O LTN será utilizado para identificar relações entre bounding boxes em imagens.*

1. Introdução

A interpretação semântica de imagens envolvendo ações representa um desafio complexo no campo da Inteligência Artificial (IA). Não se trata apenas de reconhecer objetos individuais, mas também de compreender as ações em curso e as relações espaciais entre eles. Este problema é exemplificado na análise de imagens de eventos dinâmicos, como um jogo de futebol, aonde a IA precisa identificar jogadores, a bola e entender a dinâmica do jogo, como um jogador chutando a bola [Rigues 2020].

Alguns dos principais problemas são:

- **Variação de Aparência:** A mesma ação pode parecer muito diferente dependendo de uma variedade de fatores, como o ângulo da câmera, a iluminação, o ambiente e as características individuais da pessoa realizando a ação [Santaella and Nöth 2001].

- **Oclusão:** Muitas vezes, partes importantes da ação podem estar ocultas, tornando difícil para os algoritmos de visão computacional interpretar a ação corretamente [Sabino 2023].
- **Contexto:** A interpretação correta de uma ação muitas vezes depende do contexto em que está ocorrendo. Por exemplo, a ação de uma pessoa levantando a mão pode ter significados muito diferentes dependendo se ela está em uma sala de aula, em um concerto ou praticando ioga [Sabino 2023].
- **Escalabilidade:** Com a quantidade crescente de dados visuais disponíveis, desenvolver algoritmos que possam processar e interpretar eficientemente esses dados é um desafio significativo [DSA 2021].

Nesse contexto, a abordagem da IA Neuro-Simbólica surge como uma solução promissora. Esta abordagem combina métodos simbólicos, que permitem incorporar conhecimento prévio e raciocínio lógico, com métodos baseados em aprendizado profundo, eficazes na análise de dados brutos e não estruturados, como imagens [González 2022].

Além disso, a IA Neuro-Simbólica é capaz de lidar com a incerteza e a ambiguidade inerentes à interpretação de imagens, utilizando lógica difusa para representar e raciocinar sobre informações incertas ou imprecisas [Kalatec 2022]. A lógica difusa ou lógica *Fuzzy* é a forma de lógica multivalorada, na qual os valores verdade das variáveis podem ser qualquer número real entre 0 e 1 que lida com a incerteza e a imprecisão na tomada de decisões. Ao contrário da lógica tradicional, que opera com valores booleanos (verdadeiro ou falso, 1 ou 0), a lógica difusa permite que as variáveis tenham valores de verdade em um contínuo dentro de um intervalo entre 0 e 1 [Kalatec 2022]. Sua capacidade de aprendizado e adaptação a novas situações e contextos também a torna uma escolha adequada para esse problema, usando o aprendizado profundo para absorver exemplos de treinamento e aplicar esse conhecimento a novas imagens [Kalatec 2022].

Assim, a combinação de métodos simbólicos e baseados em aprendizado profundo na IA Neuro-Simbólica oferece uma abordagem sofisticada e flexível para a interpretação semântica de imagens envolvendo ações, promovendo uma compreensão mais completa e precisa das mesmas [González 2022].

Ao aplicar a abordagem da IA Neurosimbólica para interpretar imagens de eventos dinâmicos, como um jogo de futebol, as *bounding boxes* desempenham um papel importante. Elas não apenas ajudam a identificar os objetos relevantes, mas também contribuem para entender as relações espaciais entre esses objetos e as ações em curso.

A *bounding box*, ou caixa delimitadora, é uma técnica comumente usada em visão computacional para delinear a região de interesse de um objeto em uma imagem. No contexto da interpretação semântica de imagens envolvendo ações, as *bounding boxes* são fundamentais para identificar e localizar objetos específicos, como jogadores de futebol ou a bola em um campo [Cioppa et al. 2022].

Por exemplo, ao detectar um jogador chutando a bola em uma partida de futebol, as *bounding boxes* seriam utilizadas para delimitar as áreas ocupadas pelo jogador e pela bola na imagem. Essas informações seriam então combinadas com os métodos simbólicos e baseados em aprendizado profundo da IA Neurosimbólica para compreender a dinâmica da cena, inferindo que o jogador está interagindo com a bola de uma maneira específica, como um chute [Cioppa et al. 2022].

As *bounding boxes* mostram onde estão os objetos na imagem, ajudando a IA Neurosimbólica a entender as ações de forma precisa. Isso simplifica a interpretação semântica, tornando-a mais eficaz e fácil de aplicar e entender.

2. Background

Sistemas Neuro-simbólicos de Aprendizado Os sistemas neuro-simbólicos de aprendizado representam uma abordagem híbrida na interseção entre a neurociência computacional e a inteligência artificial. Esses sistemas combinam princípios de processamento neural com métodos simbólicos para realizar tarefas de aprendizado complexas e interpretação de dados [Tenório et al. 2022]. O Capítulo 3 do livro "*Spatial-Temporal Reasoning in Symbolic Neural Network for Semantic Interpretation of Videos*" fornece uma base sólida para entender esses sistemas, explorando conceitos-chave e aplicações práticas. Alguns dos conceitos-chave são:

- **Logic Tensor Networks (LTNs):** São redes neurais simbólicas que combinam elementos de lógica simbólica e redes neurais profundas para raciocínio e aprendizado em ambientes simbólicos.
- **Spatial-Temporal Reasoning:** Refere-se à capacidade de compreender e raciocinar sobre padrões e relações espaciais e temporais em dados, como imagens e vídeos.
- **Semantic Video Interpretation:** Trata-se do processo de compreender o conteúdo semântico de vídeos, identificando objetos, ações e relações entre eles.
- **Grounded Theory in LTNs:** Envolve a construção de uma base de conhecimento sólida através da otimização dos valores de verdade das fórmulas lógicas em LTNs.
- **Activity Graphs:** São representações gráficas de eventos e relações entre objetos ao longo do tempo em vídeos, usados para modelar e entender a dinâmica das atividades.
- **Qualitative and Quantitative Spatial-Temporal Relations:** Refere-se à análise de relações espaciais e temporais tanto qualitativas quanto quantitativas entre objetos em vídeos.
- **Identification of Objects and Relations in Images and Videos:** Trata-se do processo de detecção e reconhecimento de objetos em imagens e vídeos, bem como a descrição de relações entre eles.

2.1. Fundamentos dos Sistemas Neuro-simbólicos de Aprendizado

Os sistemas neuro-simbólicos de aprendizado se baseiam na ideia de que a mente humana processa informações de maneira integrada, combinando elementos neurais e simbólicos para representar e raciocinar sobre o mundo. Essa abordagem é inspirada no funcionamento do cérebro humano, que é altamente adaptativo e eficiente na realização de tarefas cognitivas complexas [NeuroSaber 2024].

A integração de conhecimento simbólico com redes neurais artificiais representa um avanço promissor na IA explicável. As redes neurais, embora poderosas, são muitas vezes vistas como caixas-pretas devido à sua complexidade e falta de transparência [Molnar eval]. No entanto, ao incorporar conhecimento simbólico, é possível melhorar a capacidade de explicação e interpretação dos modelos de IA. Isso é evidenciado pelo uso de Redes Tensoras Lógicas (LTN), que combinam lógica difusa de primeira

ordem com aprendizado profundo para criar sistemas que não só aprendem de dados ruidosos, mas também permitem a inferência lógica e a extração de regras simbólicas [Tenório et al. 2022]. Essa abordagem híbrida aponta para uma direção onde a IA pode ser mais alinhada ao raciocínio humano, mantendo a eficiência e a capacidade de generalização das redes neurais.

- **Rede Tensor Lógico (LTN).** A LTN é um *framework* que integra Redes Neurais Artificiais e Lógica *Fuzzy* de Primeira Ordem para raciocinar sobre as propriedades da informação. Ela é capaz de identificar relações binárias entre constantes, combinando propriedades de dados de entrada, conhecimento prévio e descrições lógicas. A vantagem da LTN sobre outros frameworks é sua habilidade de combinar entrada de dados visuais com Conhecimento de Fundo (BK), evitando a aprendizagem "zero-shot", que ocorre quando um *framework* aprende sem informações prévias para guiar seu raciocínio [Tenório et al. 2022].

O aprendizado em LTN ocorre através da construção de groundings, otimizando os valores de verdade das fórmulas discutidas anteriormente. Isso resulta na criação de uma base de conhecimento LTN, também conhecida como teoria fundamentada. A teoria fundamentada é definida como uma extensão do *grounding* parcial em todos os possíveis fundamentos, de modo que todas as instâncias das cláusulas sejam satisfeitas, tornando o *grounding* satisfatório. Esse processo envolve capturar a correlação entre os atributos quantitativos de um objeto e suas propriedades relacionais [Tenório et al. 2022].

- **Integração de Redes Neurais e Lógica Simbólica.** De acordo com [Tenório et al. 2022], uma característica fundamental dos sistemas neuro-simbólicos é a integração de técnicas de redes neurais artificiais, que são capazes de aprender representações complexas de dados, com métodos de lógica simbólica. A lógica simbólica é um ramo da matemática e filosofia que lida com a representação e o raciocínio sobre informações simbólicas usando símbolos e regras formais. Essa integração permite o raciocínio formal sobre essas representações, aplicando regras de inferência para deduzir novas informações a partir do conhecimento existente [Duda 2024]. Essa combinação de redes neurais artificiais e lógica simbólica permite a criação de modelos flexíveis e poderosos capazes de lidar com dados não estruturados e realizar inferências sofisticadas.
- **Interpretabilidade em Sistemas Neuro-simbólicos.** A falta de interpretabilidade em sistemas neuro-simbólicos é uma preocupação crescente na comunidade de inteligência artificial. Muitas vezes, os usuários finais desses sistemas estão mais interessados nos resultados produzidos pela entrada e saída do sistema do que no funcionamento interno [Molnar eval]. Isso cria um problema significativo, pois a falta de compreensão sobre como o sistema chega a suas conclusões pode levar a uma confiança inadequada nos resultados e até mesmo a erros graves [O'Neil 2016].

Essa falta de interesse na interpretabilidade pode resultar em consequências negativas, como a incapacidade de corrigir problemas ou entender por que o sistema tomou uma determinada decisão. Os usuários podem simplesmente aceitar os resultados sem questionar o processo subjacente, o que pode levar a uma confiança cega e a uma dependência excessiva do sistema. Além disso, a falta de interpretabilidade pode dificultar a identificação e a correção de vieses ou falhas no sistema,

tornando-o menos transparente e responsivo às necessidades e preocupações dos usuários [O'Neil 2016].

- **Generalização em Sistemas Neuro-simbólicos.** A generalização desempenha um papel crucial na capacidade do modelo ou sistema de se adaptar e realizar de forma eficaz em dados não vistos durante o treinamento. A generalização é essencial para garantir que o sistema seja capaz de lidar com a variabilidade e complexidade dos dados encontrados nos cenários do mundo real [Vale and Barbosa 2019]. Como mencionado em [Tenório et al. 2022], a capacidade de generalização em sistemas neuro-simbólicos pode ser aprimorada através de técnicas de aprendizado que permitem ao sistema extrair características relevantes e abstrair padrões complexos dos dados de entrada.

Uma abordagem comum para melhorar a generalização em sistemas neuro-simbólicos é o uso de técnicas de regularização, como a adição de termos de penalidade ao processo de otimização durante o treinamento. Esses termos de penalidade ajudam a evitar o sobreajuste do modelo aos dados de treinamento, permitindo que ele generalize melhor para novos dados. Além disso, estratégias como a validação cruzada podem ser empregadas para avaliar a capacidade de generalização do modelo em diferentes conjuntos de dados, ajudando a identificar e mitigar problemas de sobreajuste [Tenório et al. 2022].

2.2. Principais aplicações e desafios.

Os sistemas neuro-simbólicos encontram uma ampla gama de aplicações em áreas como visão computacional e reconhecimento de padrões. Por exemplo, esses sistemas podem ser usados para interpretar imagens e vídeos, identificando objetos, reconhecendo padrões de movimento e inferindo relações espaciais e temporais entre elementos na cena [Rateke 2024].

Embora os sistemas neuro-simbólicos ofereçam muitas vantagens em termos de flexibilidade e capacidade de lidar com dados complexos, eles também enfrentam desafios significativos, incluindo:

Escalabilidade e Eficiência: A necessidade de modelos que possam escalar eficientemente para lidar com grandes volumes de dados sem comprometer a velocidade ou a precisão [Yu et al. 2023].

Integração de Conhecimento Prévio: A habilidade de incorporar conhecimento e regras pré-existentes para melhorar o aprendizado e a tomada de decisão. **Generalização e Transferência de Aprendizado:** A capacidade de generalizar aprendizados para novas situações e transferir conhecimento entre diferentes tarefas [Yu et al. 2023].

Interoperabilidade e Padrões: A criação de padrões para a interoperabilidade entre diferentes sistemas neuro-simbólicos para facilitar a colaboração e integração [Yu et al. 2023].

2.3. Trabalhos Relacionados

2.3.1. "Avanços e Desafios da Inteligência Artificial Neuro-Simbólica: Uma Perspectiva Futura"

Por Zishen Wan, Che-Kai Liu, Hanchen Yang, Chaojian Li, Haoran You, Yonggan Fu, Cheng Wan, Tushar Krishna, Yingyan (Celine) Lin, Arijit Raychowdhury. Este docu-

mento oferece uma visão abrangente dos avanços e desafios atuais da Inteligência Artificial Neuro-Simbólica (NSAI). Ele destaca a necessidade de melhorar a interpretabilidade, robustez e a capacidade de aprender com menos dados. Aqui estão algumas das estratégias discutidas:

- **Integração de Métodos:** A Inteligência Artificial Neuro-Simbólica (NSAI) combina abordagens neurais, simbólicas e probabilísticas. As abordagens neurais são baseadas em redes neurais, que são excelentes para aprender a partir de grandes quantidades de dados. As abordagens simbólicas, por outro lado, são baseadas em lógica e regras, que são excelentes para raciocínio e explicabilidade. As abordagens probabilísticas lidam com incertezas e variabilidades nos dados. Ao integrar essas abordagens, a NSAI pode melhorar a explicabilidade e robustez dos sistemas de IA.
- **Desafios Atuais:** Os sistemas de IA atuais enfrentam vários desafios. Por exemplo, eles têm trajetórias computacionais insustentáveis, o que significa que eles exigem muitos recursos computacionais para treinar e executar. Eles também têm robustez limitada, o que significa que eles podem falhar ou produzir resultados imprecisos quando expostos a situações ou dados que não foram vistos durante o treinamento. Além disso, eles carecem de explicabilidade, o que significa que é difícil entender como eles chegam às suas decisões.
- **Potencial do NSAI:** A NSAI tem um grande potencial em cenários colaborativos humano-IA. Isso é porque ela pode fornecer capacidades de raciocínio e cognição que são necessárias para trabalhar efetivamente com humanos. Por exemplo, ela pode entender e responder a perguntas complexas, fazer inferências a partir de dados limitados e explicar suas decisões de uma maneira que os humanos possam entender.
- **Futuro do NSAI:** O trabalho analisa os desafios e direções futuras do NSAI. Isso inclui a necessidade de melhorar a robustez e a explicabilidade dos sistemas de IA, bem como a necessidade de desenvolver sistemas que possam aprender com menos dados. Essas direções futuras são cruciais para o desenvolvimento contínuo da IA e podem levar a avanços significativos na área. Por exemplo, eles podem levar ao desenvolvimento de sistemas de IA que podem trabalhar efetivamente em uma variedade de cenários do mundo real, desde a medicina até a robótica.

2.3.2. "A Survey on Neural-symbolic Learning Systems"

Por Dongran Yu, Bo Yang, Dayou Liu, Hui Wang, Shirui Pan: Neste artigo, os autores realizam uma revisão abrangente sobre os avanços nos sistemas de aprendizado neuro-simbólicos. Eles discutem como esses sistemas combinam elementos neurais e simbólicos para realizar tarefas de processamento de linguagem natural de maneira interpretável e generalizável.

- **Sistemas de Aprendizado Neuro-Simbólicos:** Esses sistemas buscam combinar o melhor dos mundos neural e simbólico para criar modelos de IA que são tanto poderosos em termos de aprendizado de máquina quanto compreensíveis e lógicos como os sistemas baseados em regras. Eles são projetados para realizar tarefas complexas de processamento de linguagem natural de maneira mais interpretável e generalizável.

- **Elementos Neurais:** Referem-se às partes do sistema que são baseadas em redes neurais artificiais. Estas são estruturas computacionais inspiradas pelo funcionamento do cérebro humano e são muito boas em identificar padrões em grandes conjuntos de dados, como imagens ou texto.
- **Elementos Simbólicos:** São as partes do sistema que utilizam símbolos e regras lógicas para realizar raciocínio e inferência. Eles ajudam a tornar o processo de tomada de decisão da IA mais transparente e explicável.
- **Interpretabilidade:** A capacidade de entender e explicar as decisões tomadas por um sistema de IA. Nos sistemas neuro-simbólicos, isso é alcançado através da combinação de elementos neurais e simbólicos, o que permite que os modelos não apenas aprendam a partir de dados, mas também sigam regras lógicas compreensíveis.
- **Generalização:** A habilidade de um sistema de IA de aplicar o que aprendeu a novas situações que não foram vistas durante o treinamento. Os sistemas neuro-simbólicos são projetados para serem capazes de generalizar melhor, aplicando regras lógicas a novos dados.

3. Descrição do problema raciocínio sobre *bounding boxes* e aplicações na atualidade

As *bounding boxes*, ou caixas delimitadoras, são regiões retangulares que definem a posição e o tamanho dos objetos em uma imagem. As *bounding boxes* são amplamente utilizadas em tarefas de detecção de objetos, permitindo que modelos de aprendizado de máquina identifiquem e localizem objetos dentro de uma imagem. Eles são computacionalmente eficientes e adequados para aplicações em tempo real, o que os torna uma escolha popular em setores como veículos autônomos e vigilância por vídeo.. Sua evolução ao longo do tempo reflete o progresso contínuo na área de inteligência artificial e visão por computador. O raciocínio por trás do uso de *bounding boxes* é criar uma estrutura que permita delimitar e identificar objetos dentro de uma imagem de forma eficiente e precisa. Isso é essencial em muitas aplicações de visão computacional, onde é necessário localizar e reconhecer objetos em imagens ou vídeos [Lempitsky et al. 2009].

Inicialmente, as *bounding boxes* eram implementadas como simples retângulos que circunscreviam os objetos detectados em uma cena. Essas caixas delimitadoras eram definidas por coordenadas que especificavam os cantos superior esquerdo e inferior direito do retângulo. Embora rudimentares, essas *bounding boxes* foram um marco importante, permitindo aos algoritmos de visão computacional identificar e localizar objetos em imagens de forma mais sistemática, fornecer uma representação simples e eficiente da localização e extensão de um objeto [Ribera et al. 2019].

Com o avanço da tecnologia e o desenvolvimento de algoritmos mais sofisticados, as *bounding boxes* tornaram-se mais precisas e adaptáveis. Algoritmos de detecção de objetos baseados em redes neurais convolucionais (CNNs), como YOLO (*You Only Look Once*) e Faster R-CNN (*Region-based Convolutional Neural Network*), conseguem detectar objetos em diferentes contextos e condições de iluminação com uma precisão muito maior do que as abordagens anteriores. Esses algoritmos utilizam técnicas avançadas de processamento de imagem e aprendizado profundo para localizar e classificar objetos com rapidez e precisão [Ribera et al. 2019].

A importância das *bounding boxes* na visão computacional é evidente em uma variedade de aplicações do mundo real. Em medicina, por exemplo, as *bounding boxes* são usadas para detectar e localizar anomalias em imagens médicas, como radiografias e tomografias. Em veículos autônomos, as *bounding boxes* desempenham um papel crucial na detecção de pedestres, veículos e outros obstáculos, ajudando os sistemas de condução autônoma a tomar decisões rápidas e seguras. No reconhecimento facial, as *bounding boxes* são utilizadas para identificar e rastrear rostos em vídeos de vigilância ou em aplicativos de reconhecimento facial [Sochor et al. 2018].

Em resumo, as *bounding boxes* representam uma ferramenta essencial na detecção e localização de objetos em imagens e vídeos. Seu contínuo desenvolvimento e aprimoramento contribuem significativamente para o avanço de diversas áreas, tornando possível uma ampla gama de aplicações práticas e impulsionando o progresso contínuo na visão computacional e na inteligência artificial.

3.1. Aplicações

No contexto do futebol, o rastreamento preciso dos jogadores em campo é essencial para coletar estatísticas relevantes, como distância percorrida, posse de bola e formação da equipe. [Cioppa et al. 2022] propõe uma metodologia de baixo custo para extrair análises de futebol a partir de transmissões de TV simples, usando exclusivamente técnicas de visão computacional. O conjunto de dados SoccerNet-Tracking, criado por esse trabalho, contém 200 sequências de 30 segundos cada, representativas de cenários desafiadores do futebol, com anotações detalhadas de *bounding boxes* e IDs de trajetórias de jogadores.

Essas *bounding boxes* permitem mapear jogadores e a bola em uma visão superior estática do campo de futebol, possibilitando a dedução de insights úteis por meio da extração de análises de jogo, como posicionamento tático, movimentação dos jogadores e interações com a bola. A pesquisa contínua nessa área visa melhorar a precisão e a robustez dos métodos de rastreamento em cenários esportivos, contribuindo para o desenvolvimento de estratégias e tomadas de decisão mais informadas.

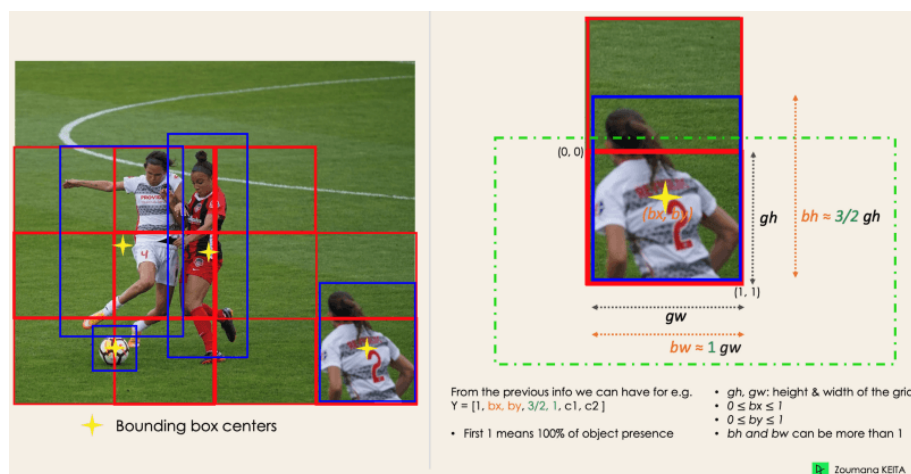


Figura 1. *Bounding Boxes* usadas nos esportes.

3.1.1. Indústria Agrícola

A utilização de *Bounding Boxes* na detecção precisa de frutos na agricultura moderna está se tornando uma prática fundamental. [Neto et al. 2019] detalha o uso de *Bounding Boxes* para identificar e contar laranjas verdes em imagens digitais de pés de laranja. Ao empregar algoritmos de detecção de objetos, como o YOLO, os agricultores podem automatizar a identificação e contagem de frutos, fornecendo uma visão detalhada da produção. Apesar dos desafios enfrentados, como variações na iluminação e oclusões por folhagem, os resultados promissores demonstram o potencial dessas técnicas. Além de facilitar a contagem, as *Bounding Boxes* abrem portas para diversas aplicações, como monitoramento de crescimento e detecção precoce de doenças, contribuindo para a eficiência e o avanço da agricultura de precisão.

No entanto, a implementação bem-sucedida dessas técnicas também requer adaptação às especificidades de cada tipo de fruto. Variações na forma e cor entre diferentes variedades exigem modelos treinados individualmente, destacando a necessidade contínua de desenvolvimento e aprimoramento. Em suma, as *Bounding Boxes* representam uma ferramenta essencial para os agricultores, capacitando-os a tomar decisões informadas e estratégicas com base em dados visuais precisos, impulsionando assim a produtividade e a eficiência na agricultura moderna [Neto et al. 2019].

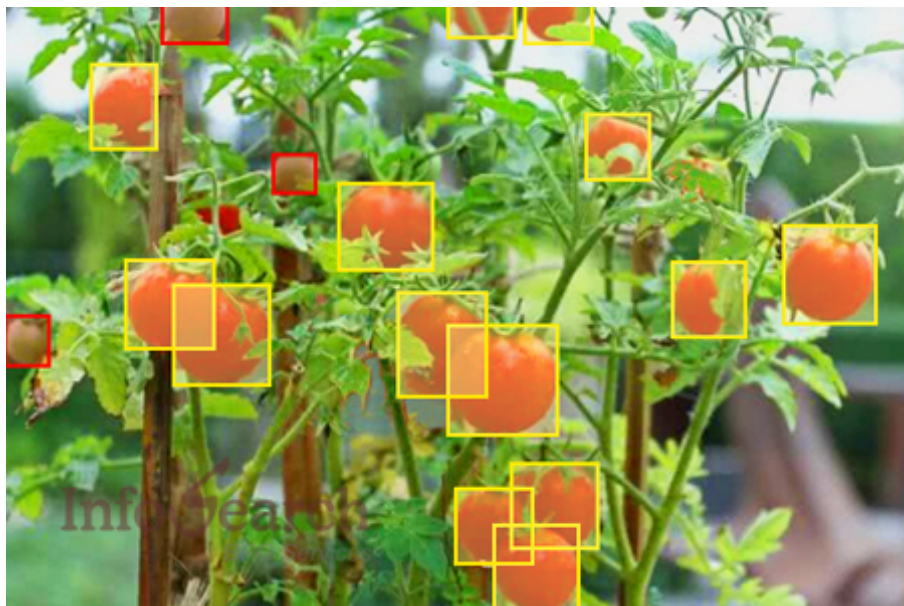


Figura 2. *Bouding Boxes* usadas na agricultura.

3.1.2. Segurança e Monitoramento de Pecuária

[Jose and Kaakinen 2022] explora a aplicação de técnicas avançadas de detecção de objetos na análise do comportamento do gado por meio de câmeras de vigilância. Métodos como YOLO e Faster R-CNN são empregados para identificar e classificar as atividades das vacas, como descanso, alimentação e permanência em pé. Essa abordagem oferece aos agricultores uma ferramenta precisa e automatizada para monitorar o comportamento

do gado, possibilitando uma gestão mais eficaz e informada.

Além disso, ele destaca os benefícios da digitalização na pecuária, visando aprimorar a eco-amizade, eficiência e lucratividade. Projetos como o AgroTeknoa estão na vanguarda dessa transformação, fornecendo acesso a novas tecnologias e conhecimentos sobre práticas agrícolas modernas. Ao introduzir máquinas, equipamentos e métodos de produção inovadores, essas iniciativas capacitam os agricultores a adotar práticas mais sustentáveis e eficientes, resultando em uma produção de gado mais eficiente e ambientalmente responsável [Jose and Kaakinen 2022].



Figura 3. Bouding Boxes usada na pecuária.

3.1.3. Detecção de navios

As caixas delimitadoras rotativas provaram ser inestimáveis em aplicações de detecção de navios. Eles são amplamente empregados na gestão pesqueira, gestão de carga, serviço de tráfego de navios e guerra naval. Ao detectar e rastrear navios com precisão, as caixas delimitadoras rotativas facilitam o gerenciamento eficiente das atividades marítimas, garantindo a conformidade com os regulamentos e melhorando a segurança e a proteção em ambientes marinhos. Ajudando na identificação de embarcações, rastreamento de rotas e áreas de pesca, e garantindo o cumprimento de quotas e regulamentações estabelecidas para a conservação de recursos marinhos [Ma et al. 2019].

Ao detectar e rastrear navios com precisão, essas caixas delimitadoras rotativas facilitam o gerenciamento eficiente das atividades marítimas em geral. Além disso, elas desempenham um papel importante na garantia da conformidade com regulamentos, promovendo a segurança marítima e proteção ambiental. Ao oferecer uma visão detalhada e em tempo real do tráfego marítimo, esses sistemas contribuem significativamente para a prevenção de acidentes, a proteção da vida humana no mar e a preservação dos ecossistemas marinhos. Assim, as caixas delimitadoras rotativas continuam a ser uma tecnologia essencial e altamente valorizada em diversas aplicações relacionadas ao transporte marítimo e à segurança naval [Ma et al. 2019].

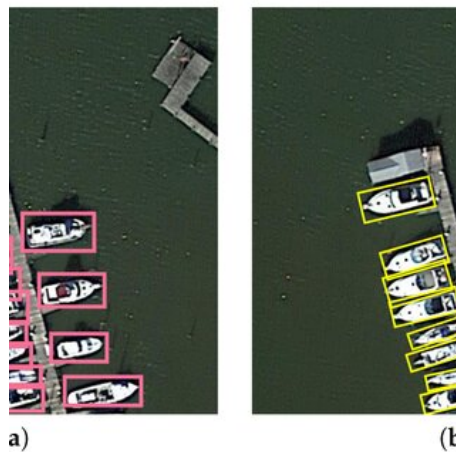


Figura 4. *Bouding Boxes* usado na detecção de navios.

4. Nossa solução

4.0.1. Codes Originais

Essa seção é responsável por administrar os códigos originais do Odaílson e por esse motivo vamos apenas dar uma breve explicada em suas partes.

- “**ltn_relations_milena.py**”: temos o código da Milena que é responsável por realizar a manipulação das caixas delimitadoras e verificar as suas relações.
- “**generate_equally_spaced_bb.py**”: há funções relacionadas à geração de caixas delimitadoras igualmente espaçadas dentro de uma caixa delimitadora fornecida.

Uma consideração importante, é que a implementação do Odaílson era feita de forma que o retorno dos predicados O, PO e D era apenas verdadeiro ou falso. Ao perceber isso, notamos que isso não fazia sentido no LTN, uma vez que o mesmo trabalha com a lógica *Fuzzy*. Ou seja, a lógica trabalha com tensores de números reais, porém os predicados retornavam apenas 0 ou 1. Portanto, decidimos refazer a implementação desses predicados.

4.1. Nossa implementação

4.1.1. Definição

Em nossa implementação, representamos os predicados a partir do princípio de áreas de interesse. Baseado no predicado, verificamos qual o percentual da área de interesse em questão em relação ao todo. Por exemplo, no predicado O, nossa área de interesse é a área em que os *bounding boxes* se sobrepõem (área verde na figura 5). Ou seja, a interseção entre eles. Daí, verificamos qual seu percentual em relação ao somatório das áreas dos *bounding boxes*.

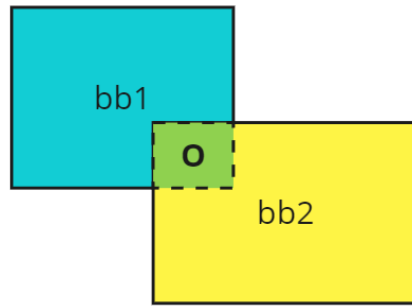


Figura 5. Exemplo do predicado O

Já no predicado D, nossa área de interesse é a área em que os *bounding boxes* não se sobrepõem (área vermelha na figura 6).

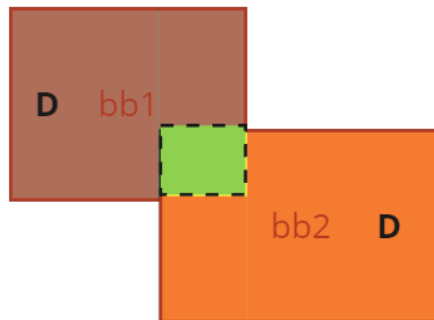


Figura 6. Exemplo do predicado D

Como os predicados D e O se complementam, definimos o predicado PO como o equilíbrio dos outros dois predicados, de forma que, quanto mais próximos os predicados estiverem entre si, maior será o valor do predicado PO. Os extremos são os casos em que o predicado D é 100% e o predicado O é 100%. Em ambos os casos, o valor de PO é 0.0. Já o maior valor que o predicado pode assumir (1.0) ocorre quando os predicados D e O são 50%. Veja a representação na figura 7.

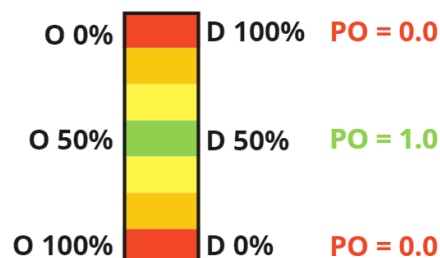


Figura 7. Representação do predicado PO

Para os predicados *right*, *left*, *above* e *below*, utilizamos a representação proposta por [Donadello 2018]. A representação utiliza o ângulo formado pelo centro dos *bounding*

boxes para o cálculo nos 4 predicados. Alguns predicados utilizam a função seno do ângulo, enquanto outros utilizam a cosseno do ângulo.

4.1.2. Código

Utilizamos diversas funções para criação dos predicados O, PO e D, porém as mais importantes são as responsáveis por retornar quais são os pontos de interseção entre os bounding boxes nos 3 casos possíveis (0, 1 e 2 pontos): **intersect_with_0_points_inside**, **intersect_with_1_points_inside** e **intersect_with_2_points_inside**. Com as funções prontas, é possível calcular qual a área da interseção entre os bounding boxes, e consequentemente descobrir qual a porcentagem da área de interesse nos predicados (retorno dos predicados O e D). No predicado PO, há apenas uma diferença: é realizado um cálculo a mais para identificar o quão próximos os predicados O e D estão.

Agora, com os predicados adaptados para a lógica *FUZZY* utilizada no LTN, convertemos todos os predicados para a forma utilizada internamente pelo LTN Torch utilizando objetos e conectivos (*AND*) LTN's.

5. Como rodar o código

Decidimos deixar o nosso código no formato de um *notebook* Google Colab para ficar mais didático e com melhor visualização. Nosso *notebook* está dividido em seções, e para rodar o código, caso o *notebook* esteja resumido como na figura 8, primeiro clique nos colchetes destacados pelos quadrados verdes. Isso fará que nosso código seja “compilado”.

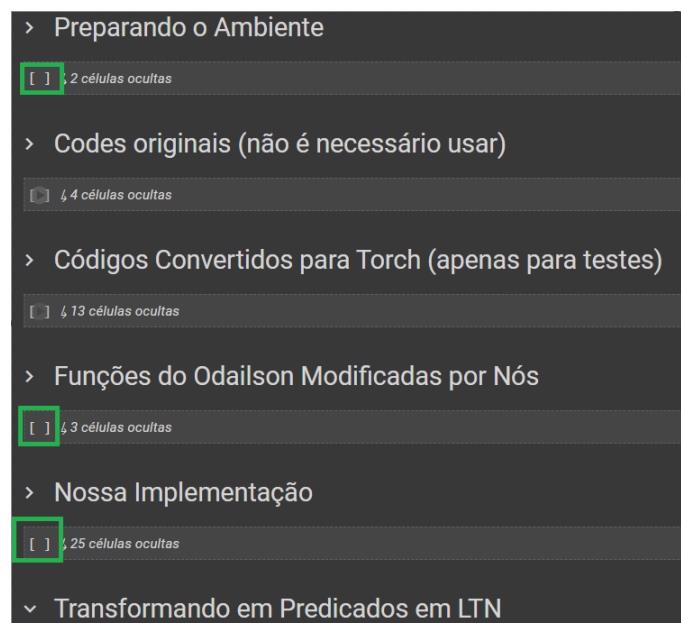


Figura 8. Seções do *Notebook*

Após isso vá para a seção **Transformando em Predicados em LTN**. Na seção, rode a célula **gerar um predicado em LTN para cada predicado no formato de função**. Agora,

defina os bounding boxes que você deseja testar na célula **definindo seus bounding boxes**. Veja na imagem 9.

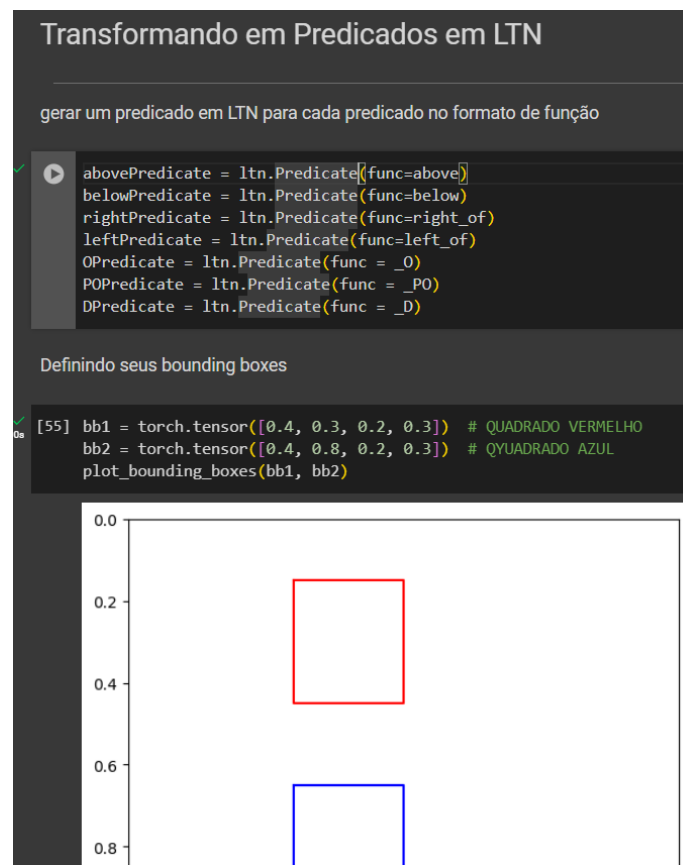


Figura 9. Seções do *Notebook*

Por fim, rode a célula **juntando predicados** e escolha um predicado que você deseja testar, basta rodar a célula do predicado. O resultado será mostrado com base nos bounding boxes que você definiu acima na saída da própria célula.



Figura 10. Seções do *Notebook*

6. Análise de seus experimentos

Para começar, temos essa célula que é responsável por gerar as *Bouding boxes* para testar os casos:

```
# TESTE
bb1 = torch.tensor([0.4, 0.3, 0.2, 0.3]) # VERMELHO
bb2 = torch.tensor([0.4, 0.8, 0.2, 0.3]) # AZUL
plot_bounding_boxes(bb1, bb2)

bb1_ltn = ltn.Constant(bb1)
bb2_ltn = ltn.Constant(bb2)
```

Figura 11. Teste geral (código) para os testes.

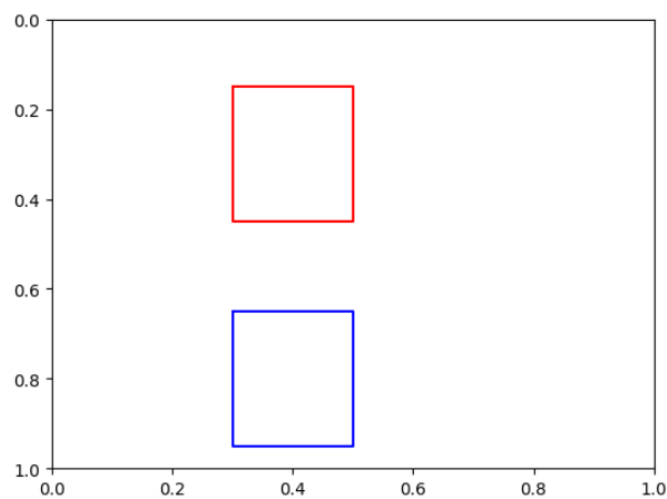


Figura 12. Teste geral (caixas geradas) para os testes.

Agora iremos rodar alguns predicados usando o teste geral acima:

Começando com o **ltnAbove**, obtivemos a seguinte saída:

```
LTNObject(value=tensor(0.7992), free_vars=[])
```

Figura 13. Saída do LTNABOVE.

Como podemos ver na saída, o valor do tensor foi 0.7992 que é um valor próximo de 1 e portanto, a *Bouding box* vermelha está acima da *Bouding box* azul.

Inclusive, trocar a ordem das *Bouding boxes* importa pois como sempre estamos comparando uma *Bouding box* com outra, estamos verificando a relação do primeiro argumento com o segundo.

Alterando a ordem, a saída 2 do **ltnabove** fica:

```
Resultado: LTNObject(value=tensor(0.), free_vars=[])
```

Figura 14. Saída do LTNABOVE.

Ou seja, a *Bouding box* azul não está acima da vermelha.

Agora vamos alterar a posição das *Bouding boxes* para deixar a vermelha na direita da azul:

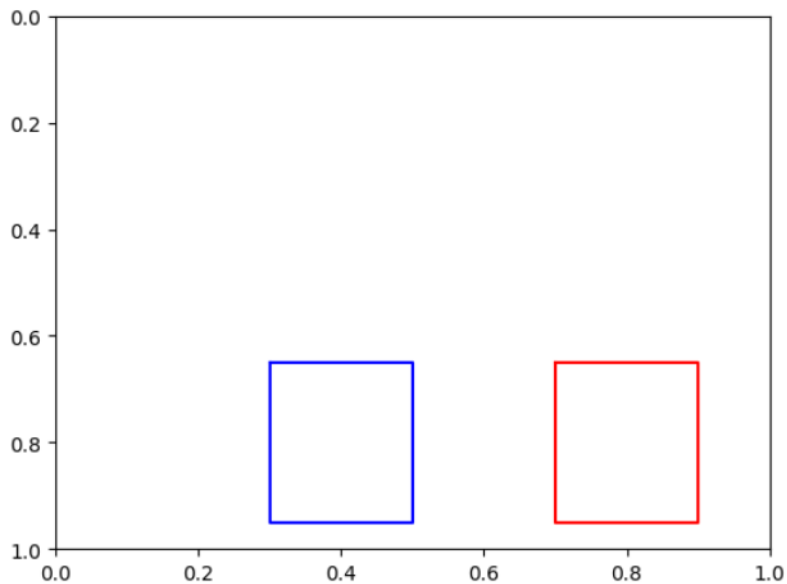


Figura 15. Teste 2.

Usando o predicado do **ltnRight**, obtivemos a seguinte saída:

```
LTNObject(value=tensor(1.), free_vars=[])
```

Figura 16. Saída do ltnRight

Portando, a saída 1 indica que a caixa vermelha está a direita da caixa azul.

Fazendo para um caso mais complicado, por exemplo, quando temos uma *Bouding box* na extremidade a esquerda da outra *Bouding box*:

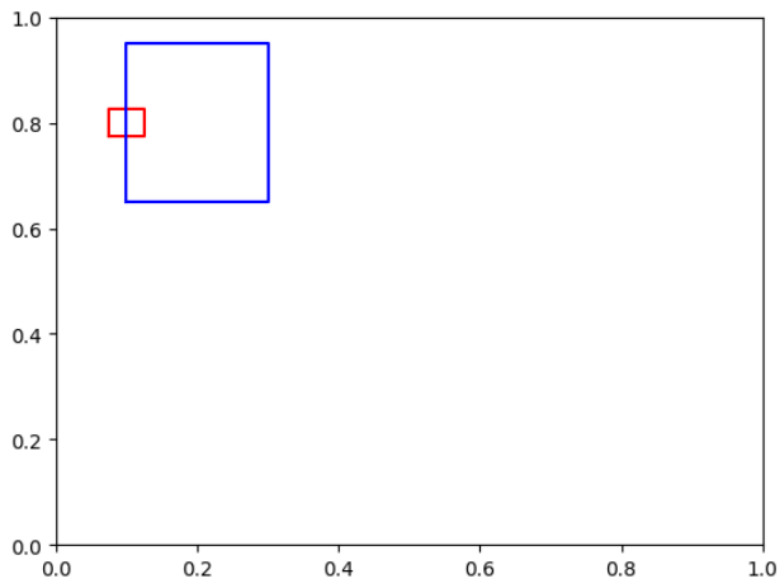


Figura 17. Teste 3

Fazendo para o predicado **ltnEdgeSideLeft** chegamos a seguinte saída:

```
tensor([[0.2000, 0.8000, 0.2000, 0.3000]]) tensor([[0.1500, 0.8000, 0.0500, 0.0500]])
tensor([[0.2000, 0.8000, 0.2000, 0.3000]]) tensor([[0.1500, 0.8000, 0.0500, 0.0500]])
tensor([[0.2000, 0.8000, 0.2000, 0.3000]]) tensor([[0.1500, 0.8000, 0.0500, 0.0500]])
totalmente nao disjuntos
LTNObject(value=tensor(0.3582), free_vars=[])
```

Figura 18. Saída do ltnEdgeSideLeft

Como pode ser visto, conseguimos o valor 0.3582 e esse valor não é 0 mas também não é muito próximo de 1. Acreditamos que chegamos neste resultado não muito bom devido à junção de predicados, já que tivemos que fazer um "and" entre o predicado *left*(leftPredicate) e o *O*(OPredicate). Ou seja, é feito uma multiplicação entre os resultados devido a implementação do *And* do ltn e pelo fato de serem valores pequenos (0 até 0.8) sempre que ocorre uma multiplicação (*and*) de predicados o valor tende a diminuir. O que gera esse valor distante e menor que 1.

A mesma lógica que mostramos aqui é seguida para cada um dos 28 predicados que podem ser testados no nosso código.

Observação: Infelizmente, tentamos iniciar o processo de criação de um modelo ltn e treinamento com base nas instruções fornecidas no github do LTNTorch, porém, não conseguimos adaptar o código fornecido para funcionar com os predicados/dataset de entrada. A última seção do notebook fornecido mostra a tentativa de adaptar o código fornecido para o nosso caso, mas falhou devido a falta de informação que é dado nos notebooks de exemplo da parte de *learning* do LTNTorch. "

Referências

- Cioppa, A., Giancola, S., Deliège, A., Kang, L., Zhou, X., Cheng, Z., Ghanem, B., and Van Droogenbroeck, M. (2022). Amateur football analytics using computer vision. *Neural Computing and Applications*, 34(8):19639–19654.
- Donadello, I. (2018). *Semantic Image Interpretation - Integration of Numerical Data and Logical Knowledge for Cognitive Vision*. PhD thesis, University of Trento.
- DSA, E. (2021). Interpretações do brasil e a temporalidade moderna: do sentimento de descompasso à crítica epistemológica.
- Duda (2024). Lógica simbólica. Acessado em 28 de Março de 2024.
- González, R. (2022). Ia neurosimbólica, todo lo que debes saber. *Big Data Magazine*.
- Jose, M. R. and Kaakinen, H. (2022). An object detection application approach to analyze cattle under camera surveillance. Centria University of Applied Sciences. Disponível em: link.
- Kalatec (2022). Lógica fuzzy: o que é, como funciona e mais. *Kalatec Blog*.
- Lempitsky, V., Kohli, P., Rother, C., and Sharp, T. (2009). Image segmentation with a bounding box prior. In *2009 IEEE 12th international conference on computer vision*, pages 277–284. IEEE.
- Ma, J., Zhou, Z., Wang, B., Zong, H., and Wu, F. (2019). Ship detection in optical satellite images via directional bounding boxes based on ship center and orientation prediction. *Remote Sensing*, 11(18):2173.
- Molnar, C. (Year of retrieval). Interpretable machine learning.
- Neto, J. C., Ternes, S., de Souza, K. X. S., Yano, I. H., and Queiros, L. R. (2019). Uso de redes neurais convolucionais para detecção de laranjas no campo. *Embrapa Informática Agropecuária*.
- NeuroSaber, I. (2024). Mecanismos neurológicos da aprendizagem. <https://iesneurosaber.com.br/mecanismos-neurologicos-da-aprendizagem-como-o-cerebro-aprende/>. Acessado em 2024, Instituto NeuroSaber para a vida).
- O’Neil, C. (2016). *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown.
- Rateke, T. (2024). Técnicas sub-simbólicas: Redes neurais. <https://lapix.ufsc.br/ensino/reconhecimento-de-padroes/tecnicas-sub-simbolicas-redes-neurais/>. Acessado em 2024, Laboratório de Pesquisas em Imagem e Som (LAPIX), Universidade Federal de Santa Catarina (UFSC).
- Ribera, J., Guera, D., Chen, Y., and Delp, E. J. (2019). Locating objects without bounding boxes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6489.
- Rigues, R. (2020). Nova ia da microsoft descreve imagens tão bem quanto os humanos. *Olhar Digital*.

- Sabino, K. (2023). Leitura de imagens: passo a passo da interpretação.
- Santaella, L. and Nöth, W. (2001). Teorias de imagem e texto. A semelhança entre palavra e imagem, a relação texto-imagem, e a construção do significado na teoria da comunicação multimodal.
- Sochor, J., Špaňhel, J., and Herout, A. (2018). Boxcars: Improving fine-grained recognition of vehicles using 3-d bounding boxes in traffic surveillance. *IEEE transactions on intelligent transportation systems*, 20(1):97–108.
- Tenório, M. R. et al. (2022). Spatial-temporal reasoning in symbolic neural network for semantic interpretation of videos.
- Vale, I. and Barbosa, A. (2019). Pensamento algébrico: contributo da visualização na construção da generalização. *Educação Matemática Pesquisa*, 21(3):398–418.
- Yu, D., Yang, B., Liu, D., Wang, H., and Pan, S. (2023). A survey on neural-symbolic learning systems. *Neural Networks*, 166:105–126.