INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE GRADUAÇÂO EM ENGENHARIA MECATRÔNICA

GERO GUSTAVO MULLER

DESENVOLVIMENTO DE UM SISTEMA WEB PARA ANOTAÇÃO DE IMAGENS VOLTADO ÀS REDES NEURAIS ARTIFICIAIS PROFUNDAS PARA DETECÇÃO DE OBJETOS

FLORIANÓPOLIS, 2025.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA - CÂMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE METAL-MECÂNICA
CURSO DE GRADUAÇÂO EM ENGENHARIA MECATRÔNICA

GERO GUSTAVO MULLER

DESENVOLVIMENTO DE UM SISTEMA WEB PARA ANOTAÇÃO DE IMAGENS VOLTADO ÀS REDES NEURAIS ARTIFICIAIS PROFUNDAS PARA DETECÇÃO DE OBJETOS

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Engenheiro Mecatrônico.

Orientador: Prof. Maurício Edgar Stivanello, Dr. Eng.

FLORIANÓPOLIS, 2025.

Ficha de identificação da obra elaborada pelo autor.

Müller, Gero Gustavo Müller
DESENVOLVIMENTO DE UM SISTEMA WEB PARA ANOTAÇÃO DE IMAGENS VOLTADO ÀS REDES NEURAIS ARTIFICIAIS PROFUNDAS PARA DETECÇÃO DE OBJETOS / Gero Gustavo Müller Müller; orientação de Maurício Edgar Stivanello Stivanello. - Florianópolis, SC, 2025.
63 p.
Trabalho de Conclusão de Curso (TCC) - Instituto Federal de Santa Catarina, Câmpus Florianópolis. Bacharelado em Engenharia Mecatrônica. Departamento Acadêmico de Metal Mecânica.
Inclui Referências.

1. Anotação de imagens. 2. Aplicação web. 3. Detecção de objetos. 4. Sistemas de visão. 5. Redes Neurais Profundas. I. Stivanello, Maurício Edgar Stivanello. II. Instituto Federal de Santa Catarina. III. DESENVOLVIMENTO DE UM SISTEMA WEB PARA ANOTAÇÃO DE IMAGENS VOLTADO ÀS REDES NEURAIS ARTIFICIAIS PROFUNDAS PARA DETECÇÃO

DESENVOLVIMENTO DE UM SISTEMA WEB PARA ANOTAÇÃO DE IMAGENS VOLTADO ÀS REDES NEURAIS ARTIFICIAIS PROFUNDAS PARA DETECÇÃO DE OBJETOS

GERO GUSTAVO MÜLLER

Este trabalho foi julgado adequado para obtenção do título de Engenheiro Mecatrônico em 2025 e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Mecatrônica do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 27 de Fevereiro de 2025.

Banca Examinadora:

gov.br Documento assinado digitalmente
MAURICIO EDGAR STIVANELLO
Data: 31/03/2025 19:21:39-0300
Verifique em https://validar.iti.gov.br

Prof. Maurício Edgar Stivanello, Dr. Eng.

gov.br Documento assinado digitalmente
DELCINO PICININ JUNIOR
Data: 31/03/2025 13:04:11-0300
Verifique em https://validar.iti.gov.br

Prof. Delcino Picinin Júnior, Dr. Eng.

gov.br Documento assinado digitalmente
RAIMUNDO RICARDO MATOS DA CUNHA
Data: 31/03/2025 13:18:56-0300
Verifique em https://validar.iti.gov.br

Prof. Raimundo Ricardo Matos da Cunha, Dr. Eng.

AGRADECIMENTOS

Ao Instituto Federal de Santa Catarina – campus Florianópolis, ao Departamento Acadêmico de Metal-Mecânica, e ao seu corpo docente e administrativo, pelo ensino de excelência e estrutura oferecida ao longo da minha formação.

Agradecimento especial ao professor Dr. Eng. Maurício Edgar Stivanello, orientador deste trabalho, pela oportunidade de explorar o tema proposto e pelo acompanhamento primoroso ao longo de toda a pesquisa, sempre contribuindo com seu conhecimento e direcionamento.

À minha família, pelo apoio incondicional, incentivo e confiança, que me permitiram seguir sempre em frente.

RESUMO

A anotação de imagens tem se mostrado fundamental no avanço dos sistemas computacionais integrados à inteligência artificial, especialmente em visão computacional. Esses sistemas evoluíram notavelmente, com destaque para a detecção de objetos. As Redes Neurais Profundas (RNPs) desempenham papel crucial nesse avanço, graças à sua habilidade em aprender representações de padrões visuais em imagens. No entanto, para alcançar a eficácia desejada, as RNPs precisam ser treinadas com bases de dados robustas. A anotação envolve a identificação detalhada de objetos em imagens, delineando suas características e classes. Este trabalho teve como objetivo principal o desenvolvimento de uma aplicação web para facilitar a anotação de imagens, permitindo aos usuários demarcar regiões específicas em imagens. A solução foi construída com tecnologias web modernas, visando compatibilidade, escalabilidade e uma interface intuitiva que otimiza a experiência do usuário. Além disso, foi implementada uma estrutura para integração, persistência e gestão dos dados anotados, facilitando o manejo de conjuntos de dados cruciais para o treinamento eficaz de RNPs.

Palavras-chave:

Palavras-chave: Anotação de imagens. Aplicação web. Detecção de objetos. Redes Neurais Profundas (RNPs). Sistemas de visão.

ABSTRACT

Image annotation has proven to be fundamental in the advancement of computational systems integrated with artificial intelligence, especially in computer vision systems. These systems have evolved remarkably, with a highlight on object detection. Deep Neural Networks (DNNs) play a crucial role in this advancement, thanks to their ability to learn representations of visual patterns in images. However, to achieve the desired efficacy, DNNs needed to be trained with robust databases. Annotation involves the detailed identification of objects in images, delineating their characteristics and classes. This work aimed to develop a web application to facilitate image annotation. The application allows users to demarcate specific regions in images for the training of DNNs in object detection. The solution was built using modern web technologies, proposing compatibility and scalability. An intuitive interface was created to optimize the user's annotation experience. The flexibility of this approach allows for future expansions and improvements, including the addition of new functionalities or integration with advanced image detection algorithms. In addition to the annotation interface, a structure for the integration, persistence, and management of annotated data was developed. This facilitates the handling of annotated data sets, crucial in the training of DNNs. The main goal of this work was to provide a web tool dedicated to image annotation, contributing to the advancement of future research in Deep Neural Networks. This tool enhanced the creation, manipulation, and management of annotated data sets, essential for pattern recognition and effective training of DNNs.

Keywords: Deep Neural Networks (DNNs). Image annotation. Object detection. Vision systems. Web application.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ADAM	*Adaptive Moment Estimation*
ASIC	*Application Specific Integrated Circuits*
COCO	*Common Objects in Context*
CVAT	*Computer Vision Annotation Tool*
CNN	*Convolutional Neural Network*
DNN	*Deep Neural Network* (Rede Neural Profunda)
GPU	*Graphics Processing Unit*
HOG	*Histogram of Oriented Gradients*
IFSC	*Instituto Federal de Santa Catarina*
ILSVRC	*ImageNet Large Scale Visual Recognition Challenge*
JSON	*JavaScript Object Notation*
RNC	*Rede Neural Convolucional*
RNP	*Rede Neural Profunda*
ReLU	*Rectified Linear Unit*
SGD	*Stochastic Gradient Descent*
SIFT	*Scale-Invariant Feature Transform*
SVM	*Support Vector Machines*
TPU	*Tensor Processing Unit*
VOC	*Visual Object Classes*
YOLO	*You Only Look Once*

SUMÁRIO

1 INTRODUÇÃO

O processamento de imagens tem impulsionado avanços significativos na inteligência artificial, especialmente na visão computacional. Essa área permite que máquinas e dispositivos interpretem e interajam com o mundo visual de maneira sofisticada, ampliando as possibilidades de automação, análise de dados e reconhecimento de padrões (SZELISKI, 2010).

Uma das aplicações mais relevantes da visão computacional é a detecção de objetos, técnica que envolve a identificação e localização de elementos específicos em imagens. Esse processo é fundamental em diversas áreas, como navegação autônoma de veículos, sistemas de monitoramento, diagnóstico por imagem na medicina e robótica industrial. As Redes Neurais Profundas (RNPs) destacam-se nesse campo, pois modelos de aprendizado profundo aprimoram continuamente sua capacidade de reconhecer e classificar objetos com base em representações visuais (LECUN; BENGIO; HINTON, 2015).

O sucesso dos modelos de detecção de objetos depende diretamente da qualidade e da quantidade de dados utilizados no treinamento. Para isso, a anotação de imagens desempenha um papel essencial, pois consiste em marcar manualmente objetos dentro das imagens, criando um conjunto de dados estruturado para que as redes neurais possam aprender com maior precisão (RUSSAKOVSKY *et al.*, 2014).

Entretanto, o processo de anotação pode ser trabalhoso, demandando ferramentas eficientes para facilitar essa tarefa. Diante desse desafio, este trabalho propõe o desenvolvimento de um sistema web para anotação de imagens, permitindo que usuários realizem marcações de forma intuitiva e eficiente. O sistema foi projetado para ser simples e acessível, possibilitando a exportação dos dados anotados no formato YOLO, amplamente utilizado em redes neurais para detecção de objetos.

O propósito deste trabalho é facilitar a criação, manipulação e gerenciamento de conjuntos de dados para treinamento de modelos de visão computacional. Dessa forma, busca-se contribuir para o avanço da detecção de objetos, fornecendo uma ferramenta prática para pesquisadores e desenvolvedores.

1.1 Justificativa

A anotação de imagens desempenha um papel essencial na preparação de conjuntos de dados para modelos de visão computacional. Definir atributos apropriados para cada imagem permite identificar objetos com precisão, influenciando diretamente a eficácia dos algoritmos de detecção. Esse processo é fundamental para Redes Neurais Profundas (RNPs), que utilizam grandes volumes de imagens anotadas para aprimorar sua capacidade de reconhecimento e classificação.

Para que uma rede neural seja treinada de forma eficaz, é necessário um conjunto de dados substancial, contendo imagens rotuladas com precisão. Embora existam diversas ferramentas para anotação manual, muitas delas apresentam interfaces pouco intuitivas, são pagas ou exigem configurações complexas, dificultando o acesso para pesquisadores e desenvolvedores. (SAGER; ZSCHECH, 2021).

Diante desse cenário, justifica-se a criação de uma ferramenta de acesso livre e facilitado para a anotação de imagens. Este trabalho propõe o desenvolvimento de um sistema web intuitivo, projetado para agilizar a anotação e facilitar a exportação de dados no formato YOLO, um dos mais utilizados para treinamento de modelos de detecção de objetos.

O uso de uma aplicação web oferece vantagens significativas, como acessibilidade, escalabilidade e facilidade de uso, eliminando a dependência de softwares pagos ou que exigem configurações complexas. Além disso, a possibilidade de colaboração entre usuários em um ambiente online amplia a flexibilidade do processo de anotação, permitindo um fluxo de trabalho mais eficiente e acessível.

Portanto, este trabalho propõe o desenvolvimento de uma solução web para anotação de imagens, permitindo que pesquisadores e desenvolvedores criem conjuntos de dados de forma mais eficiente. Ao tornar o processo de anotação mais acessível e organizado, a ferramenta contribui diretamente para o avanço da visão computacional e da detecção de objetos, otimizando um processo essencial para o treinamento de modelos baseados em inteligência artificial.

1.2 Definição do Problema

Este trabalho apresenta o desenvolvimento de uma solução acessível para anotação de imagens, com foco no contexto acadêmico. Embora existam diversas soluções comerciais avançadas, muitas possuem restrições quanto à customização e usabilidade, além de frequentemente exigirem planos pagos para acesso a funcionalidades essenciais, o que pode limitar seu uso em pesquisas acadêmicas e projetos independentes.

Dessa forma, este projeto propõe uma plataforma que não apenas simplifique o processo de anotação de imagens, mas também possibilite sua evolução conforme as demandas dos usuários. Criar uma solução adaptável permite que pesquisadores e desenvolvedores utilizem a ferramenta de forma flexível na geração de conjuntos de dados, sem a necessidade de infraestrutura complexa ou licenças pagas, contribuindo assim para a expansão do projeto e para o avanço dos sistemas de visão computacional.

1.3 Objetivos

A proposta deste trabalho é desenvolver uma ferramenta web para anotação de imagens, facilitando a criação de conjuntos de dados utilizados em visão computacional.

1.3.1 Objetivo Geral

O desenvolvimento de uma interface web acessível para anotação de imagens visa atender às demandas do meio acadêmico e da pesquisa em visão computacional. A ferramenta será intuitiva e compatível com formatos amplamente utilizados, permitindo sua integração com trabalhos futuros.

1.3.2 Objetivos Específicos

a) Estudar os formatos de anotação de imagens existentes, avaliando suas vantagens, limitações e aplicabilidade em diferentes contextos;

b) Selecionar tecnologias web adequadas para o desenvolvimento da ferramenta, garantindo eficiência e compatibilidade;

c) Implementar uma ferramenta web para anotação de imagens, compatível com os formatos mais utilizados e focada em usabilidade;

d) Monitorar e avaliar o desempenho da ferramenta após sua implementação, garantindo funcionalidade;

e) Validar a ferramenta em ambiente controlado, testando sua usabilidade e verificando a exportação das anotações no formato adequado;

f) Disponibilizar a ferramenta e a documentação em um repositório público, promovendo o compartilhamento e colaboração;

g) Documentar o desenvolvimento do sistema, registrando as decisões técnicas e lições aprendidas.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a base conceitual necessária para a compreensão e o desenvolvimento deste estudo. Esta seção aborda os principais conceitos, teorias e pesquisas relacionadas ao campo da visão computacional, com foco na anotação de imagens e nos métodos utilizados para preparar datasets para modelos de detecção de objetos.

A anotação de imagens desempenha um papel fundamental na eficiência e precisão de algoritmos de visão computacional. Modelos modernos, como YOLO e outras abordagens baseadas em aprendizado profundo, dependem de grandes volumes de dados anotados para aprender a identificar e classificar objetos em imagens. Assim, a qualidade das anotações impacta diretamente o desempenho do modelo, tornando essencial a adoção de técnicas e ferramentas apropriadas para esse processo.

Além disso, a seção explora as tecnologias empregadas no processamento e análise de imagens, destacando a evolução dos métodos de detecção de objetos, as abordagens utilizadas na anotação e os desafios enfrentados na construção de datasets robustos. Dessa forma, a fundamentação teórica servirá como suporte para o desenvolvimento da ferramenta de anotação de imagens proposta neste trabalho.

2.1 Visão Computacional e Detecção de Objetos

A visão computacional é um dos campos mais dinâmicos da inteligência artificial, responsável por capacitar máquinas a interpretar e processar informações visuais de forma semelhante à percepção humana. Esse campo vai além da captura de imagens, abrangendo a extração e análise de características visuais para reconhecer padrões, identificar objetos e compreender contextos complexos.

Atualmente, os modelos modernos de detecção de objetos, como YOLO (You Only Look Once), Faster R-CNN e SSD (Single Shot MultiBox Detector), exigem grandes volumes de dados anotados para seu treinamento. A qualidade da anotação impacta diretamente a precisão e robustez desses sistemas, tornando essencial a criação de datasets bem estruturados, com marcações corretas e padronizadas.(SZELISKI, 2010).

Dessa forma, esta seção explora a evolução da visão computacional, com foco na detecção de objetos e na importância da anotação de imagens como etapa crítica para o desenvolvimento e aprimoramento de modelos de inteligência artificial.(FORSYTH; PONCE, 2002).

2.1.1 Desenvolvimento Histórico

A visão computacional teve suas origens entre as décadas de 1950 e 1960, quando os primeiros estudos começaram a explorar o reconhecimento de padrões

visuais. A motivação inicial era permitir que computadores interpretassem imagens de maneira semelhante à percepção humana. Durante esse período, pesquisadores desenvolveram técnicas básicas para extração de bordas e segmentação de imagens, estabelecendo os fundamentos para a evolução posterior da área.

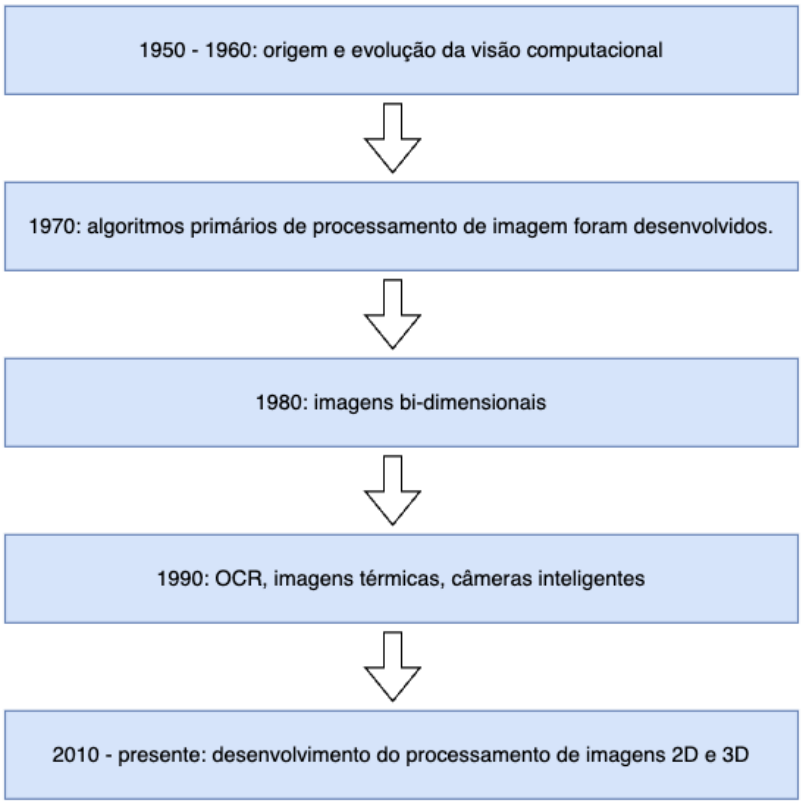
Na década de 1970, os primeiros algoritmos de processamento de imagens foram desenvolvidos, permitindo a análise digital de imagens por computadores. Avanços matemáticos e estatísticos impulsionaram o desenvolvimento de técnicas como a Transformada de Hough, introduzida em 1962 por Paul Hough e aprimorada ao longo da década seguinte. Essa técnica possibilitou a detecção de formas geométricas em imagens digitais, contribuindo para a evolução do reconhecimento de padrões e segmentação de imagens.

Na década de 1980, os avanços na capacidade computacional possibilitaram o uso de imagens bidimensionais para análise visual. Durante esse período, houve uma expansão significativa no uso de algoritmos para aprimoramento de imagens, incluindo detecção de bordas e filtragem espacial. Isso estabeleceu bases importantes para o desenvolvimento de técnicas mais avançadas de reconhecimento de padrões.

A década de 1990 marcou um grande salto no desenvolvimento da visão computacional, com a introdução de novas técnicas como OCR (Optical Character Recognition), imagens térmicas e câmeras inteligentes. Além disso, as redes neurais começaram a ganhar destaque, com a introdução de modelos como LeNet-5, desenvolvido por Yann LeCun em 1998. Esse modelo demonstrou a viabilidade do uso de Redes Neurais Convolucionais (CNNs) para tarefas como reconhecimento de caracteres, abrindo caminho para aplicações mais complexas na área.

A partir de 2010, o avanço das redes neurais profundas (deep learning) transformou a visão computacional, possibilitando o processamento eficiente de imagens 2D e 3D. Modelos como AlexNet, desenvolvido por Krizhevsky em 2012, demonstraram o potencial das CNNs para análise visual em larga escala. O crescimento da disponibilidade de grandes conjuntos de dados (big data) e o aumento da capacidade computacional com o uso de GPUs permitiram o desenvolvimento de modelos cada vez mais sofisticados. Atualmente, técnicas como YOLO (You Only Look Once) e Faster R-CNN são amplamente utilizadas para detecção de objetos em tempo real, destacando a importância da anotação precisa de imagens para o treinamento de modelos robustos. A Figura 1 apresenta uma linha do tempo ilustrando a evolução da visão computacional, destacando os principais avanços ao longo das décadas. (KRIZHEVSKY SUTSKEVER, 2012).

Figura 1 – História da Visão Computacional no Tempo



Fonte: Do autor (2023).

Nos últimos anos, a evolução da visão computacional foi impulsionada pelo desenvolvimento de TPUs (Tensor Processing Units) e frameworks como TensorFlow e PyTorch, que facilitaram a implementação e treinamento de modelos avançados. Além disso, arquiteturas modernas, como o YOLO (You Only Look Once) e Vision Transformers (ViTs), continuam ampliando as aplicações da visão computacional para novas áreas, como veículos autônomos, diagnóstico médico e reconhecimento de gestos (FORSYTH; PONCE, 2002).

2.1.2 Técnicas Fundamentais da Visão Computacional

A visão computacional depende de uma série de técnicas para processar, analisar e interpretar imagens digitais. Esses métodos permitem desde a manipulação básica de pixels até processos avançados de aprendizado profundo para reconhecimento de padrões complexos. Esta seção aborda dois pilares essenciais da visão computacional: o processamento digital de imagens e a detecção de objetos, destacando tanto métodos clássicos quanto abordagens modernas baseadas em redes neurais.

2.1.2.1 Processamento Digital de Imagens

O processamento digital de imagens é uma etapa fundamental para aprimorar a qualidade das imagens e facilitar a extração de características relevantes. Esse processamento envolve técnicas como suavização, realce de bordas e filtragem, que auxiliam na remoção de ruídos, na detecção de contornos e na melhoria da precisão de modelos de visão computacional.

- Realce de Bordas e Suavização: O realce de bordas é uma técnica essencial na análise de imagens, pois permite destacar as transições entre diferentes regiões de uma imagem, facilitando a identificação de objetos e padrões. Para isso, são utilizados operadores de detecção de bordas, como Sobel e Canny (GONZALEZ; WOODS, 2018).

- Filtro Sobel: Calcula a derivada da imagem para identificar regiões de transição acentuada entre pixels. É amplamente utilizado para destacar contornos em imagens.

- Filtro Gaussian Blur: Suaviza a imagem reduzindo ruídos e detalhes irrelevantes, preparando-a para a extração de bordas e características.

- Detector de Canny: Método mais avançado que envolve suavização, cálculo do gradiente, supressão de não máximos e limiares duplos para destacar bordas mais relevantes.

A Figura 2 apresenta um exemplo da aplicação do filtro de Sobel para detecção de bordas.

Figura 2 – Aplicação do filtro de Sobel na Detecção de Bordas

Fonte: Do autor (2024).

A suavização e o realce de bordas são fundamentais para a segmentação de imagens, um processo que divide uma imagem em regiões significativas, destacando os objetos de interesse para posterior análise.

2.1.2.2 Detecção de Objetos

A detecção de objetos é uma das tarefas mais estudadas na visão computacional. Seu objetivo é identificar e localizar objetos específicos dentro de uma imagem ou sequência de vídeo. Para isso, diferentes métodos foram desenvolvidos ao longo do tempo, evoluindo de abordagens baseadas na extração manual de características para redes neurais profundas treinadas com grandes bases de dados.

Historicamente, os primeiros métodos de detecção de objetos baseavam-se em características extraídas manualmente das imagens, como bordas, texturas e pontos de interesse. No entanto, essas abordagens apresentavam limitações em termos de robustez e generalização, especialmente em cenários complexos com variações de iluminação, perspectiva e oclusão. Com o avanço das redes neurais profundas, tornou-se possível treinar modelos capazes de aprender automaticamente representações mais abstratas e robustas, melhorando significativamente a precisão e a eficiência na detecção de objetos.

2.1.3 Métodos Clássicos de Detecção de Objetos

Antes da ascensão das redes neurais profundas, a detecção de objetos dependia de algoritmos tradicionais baseados na extração manual de características. Entre os principais métodos clássicos, destacam-se:

- Haar Cascades: desenvolvido para o reconhecimento facial, o método Haar Cascades utiliza cascatas de classificadores baseadas em características extraídas de regiões da imagem. Ele emprega janelas deslizantes para detectar padrões específicos, tornando-se um dos primeiros métodos eficientes de detecção em tempo real. Entretanto, essa técnica apresenta limitações quando há variações significativas de iluminação, ângulo e escala, resultando em altas taxas de falsos positivos e negativos (VIOLA; JONES, 2001).

- SIFT (Scale-Invariant Feature Transform): o algoritmo SIFT permite detectar e descrever pontos-chave de uma imagem de maneira robusta a mudanças de escala e rotação. Ele identifica características invariantes, facilitando a correspondência de objetos em diferentes imagens. Esse método foi amplamente utilizado em aplicações de reconhecimento de imagens, embora seja computacionalmente mais custoso em comparação com outras técnicas (LOWE, 2004).

- HOG (Histogram of Oriented Gradients): o método HOG analisa gradientes de intensidade e direções de borda para extrair padrões visuais característicos. É particularmente eficaz para o reconhecimento de pedestres, pois destaca contornos e formas distintas, permitindo que modelos de aprendizado de máquina,

como SVM (Support Vector Machines), classifiquem as imagens com alta precisão (DALAL; TRIGGS, 2005).

Esses métodos clássicos foram fundamentais no desenvolvimento inicial da visão computacional e ainda são utilizados em cenários específicos. No entanto, suas limitações em lidar com variações complexas de iluminação, escala e perspectiva impulsionaram a adoção de técnicas baseadas em aprendizado profundo, que oferecem maior flexibilidade e precisão.

2.1.4 Métodos Baseados em Aprendizado Profundo

Com o avanço do aprendizado profundo, a detecção de objetos evoluiu significativamente, deixando para trás as técnicas baseadas na extração manual de características. Em vez de depender de descritores como HOG e SIFT, os modelos modernos utilizam Redes Neurais Convolucionais (RNCs) para aprender automaticamente padrões a partir de grandes bases de dados anotadas. Isso permitiu ganhos expressivos em precisão, robustez e escalabilidade, tornando a detecção de objetos mais eficaz mesmo em cenários desafiadores, como variações de iluminação, oclusão e mudanças de escala.

Atualmente, três das abordagens mais populares para detecção de objetos baseadas em aprendizado profundo são:

- YOLO (You Only Look Once) – Detecção em Tempo Real: o YOLO revolucionou a detecção de objetos ao introduzir um modelo unificado, capaz de identificar múltiplos objetos em uma única passagem pela rede neural. Diferente de métodos anteriores, que dividem a detecção em múltiplas etapas, o YOLO realiza a previsão de classes e localizações de objetos simultaneamente, tornando-se extremamente rápido e adequado para aplicações em tempo real, como vigilância, veículos autônomos e robótica (REDMON *et al.*, 2016).

- Faster R-CNN – Alta Precisão e Segmentação Detalhada: o Faster R-CNN aprimorou os métodos baseados em proposta de regiões (R-CNN e Fast R-CNN) ao introduzir a Rede de Propostas Regionais, que gera áreas candidatas para a detecção de objetos de forma mais eficiente. Esse modelo atinge altíssima precisão, sendo amplamente utilizado em aplicações que exigem resultados detalhados, como análise médica e inspeção industrial. No entanto, seu alto custo computacional o torna menos viável para uso em tempo real (REN *et al.*, 2015).

- SSD (Single Shot MultiBox Detector) – Balanceando Velocidade e Precisão: o SSD combina aspectos do YOLO e do Faster R-CNN, utilizando múltiplas camadas convolucionais para detectar objetos em diferentes escalas dentro da mesma

imagem. Isso melhora a precisão para objetos menores sem comprometer a velocidade, tornando-o uma alternativa interessante para aplicações que exigem um equilíbrio entre rapidez e acurácia, como drones autônomos e sistemas de segurança (LIU *et al.*, 2016).

Essas técnicas baseadas em aprendizado profundo trouxeram uma evolução significativa na visão computacional, permitindo que modelos de detecção operem com maior eficiência, precisão e flexibilidade. A escolha do modelo depende do contexto da aplicação: YOLO para tarefas em tempo real, Faster R-CNN para aplicações que priorizam precisão e SSD para cenários que necessitam de um meio-termo entre desempenho e detalhamento.

2.2 Redes Neurais Profundas

A evolução da visão computacional está diretamente ligada ao avanço das redes neurais profundas (RNPs). Essas arquiteturas de aprendizado de máquina foram inspiradas no funcionamento do cérebro humano e são responsáveis por avanços significativos em diversas aplicações, incluindo reconhecimento facial, tradução automática, diagnóstico médico e, especialmente, na detecção e classificação de objetos em imagens.

A principal característica das redes neurais profundas é a capacidade de aprender representações complexas dos dados por meio de múltiplas camadas de processamento. Diferente de abordagens anteriores, que dependiam de técnicas manuais de extração de características, as redes profundas conseguem identificar padrões automaticamente, reduzindo a necessidade de intervenção humana no pré-processamento das imagens. No entanto, estas redes ainda dependem fundamentalmente de dados de alta qualidade, especialmente de imagens corretamente anotadas para seu treinamento efetivo.

Neste contexto, as Redes Neurais Convolucionais (RNCs) surgiram como um avanço crucial para a visão computacional, permitindo que modelos como YOLO (You Only Look Once), Faster R-CNN e SSD atingissem alto desempenho na detecção de objetos. Além disso, a evolução do hardware, como GPUs (Graphics Processing Units) e TPUs (Tensor Processing Units), possibilitou o treinamento de modelos cada vez mais robustos e eficientes, acelerando o desenvolvimento de novas aplicações baseadas em redes neurais profundas.

Nesta seção, serão abordados os principais conceitos relacionados às redes neurais profundas, suas arquiteturas mais utilizadas e os impactos tecnológicos que tornaram possível sua aplicação em larga escala, destacando a importância das anotações de imagens para o treinamento eficaz desses sistemas.

2.2.1 Arquitetura das Redes Neurais

As redes neurais são modelos computacionais inspirados na estrutura e no funcionamento do cérebro humano. Elas são compostas por neurônios artificiais organizados em camadas interconectadas, capazes de aprender padrões a partir de dados e realizar tarefas complexas, como reconhecimento de imagens e processamento de linguagem natural. A estrutura de uma RNA pode variar dependendo do problema a ser resolvido, mas, de forma geral, segue uma organização em três tipos principais de camadas: camada de entrada, camadas ocultas e camada de saída (GOODFELLOW; BENGIO; COURVILLE, 2016).

A camada de entrada recebe os dados brutos e os transfere para as camadas seguintes. No caso da visão computacional, por exemplo, essa camada processa os valores numéricos correspondentes aos pixels de uma imagem. As camadas ocultas são responsáveis pelo processamento interno da informação, aplicando funções matemáticas complexas para detectar padrões e extrair características relevantes dos dados. Quanto maior o número de camadas ocultas, mais profunda é a rede neural, o que possibilita a captação de padrões mais abstratos e sofisticados. Já a camada de saída gera o resultado final da rede, que pode ser uma classificação, uma predição numérica ou um conjunto de coordenadas espaciais no caso de detecção de objetos (SCHMIDHUBER, 2015).

O funcionamento das redes neurais se baseia no conceito de pesos sinápticos, que determinam a influência de um neurônio sobre o outro. Durante o treinamento, esses pesos são ajustados por meio do algoritmo de retropropagação do erro (backpropagation), que minimiza a diferença entre a saída prevista e a saída real, reduzindo o erro do modelo a cada iteração (RUMELHART; HINTON; WILLIAMS, 1986). Esse processo é conduzido pelo otimizador, um algoritmo que ajusta os pesos com base no gradiente do erro. Entre os otimizadores mais comuns estão o SGD (Stochastic Gradient Descent) e o ADAM (Adaptive Moment Estimation), amplamente utilizados no treinamento de redes neurais profundas.

Para garantir a eficiência do treinamento, as redes neurais utilizam funções de ativação, responsáveis por introduzir não linearidade ao modelo e permitir a aprendizagem de padrões complexos. Dentre as funções de ativação mais utilizadas destacam-se a ReLU (Rectified Linear Unit), que melhora a convergência dos modelos, a sigmoide, empregada em classificações binárias, e a softmax, comumente utilizada para problemas de classificação multiclasse.

Além disso, arquiteturas modernas de redes neurais são projetadas para evitar problemas como overfitting, que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento e perde a capacidade de generalização. Para mitigar esse problema, técnicas como dropout, regularização e aumento de dados (data

augmentation) são frequentemente aplicadas. (SRIVASTAVA et al., 2014).

Novas arquiteturas de redes neurais têm sido propostas para diferentes aplicações. Modelos como Redes Neurais Convolucionais (RNCs) são especialmente eficazes para análise de imagens por exemplo. A evolução desses modelos tem permitido avanços expressivos em diversas áreas da engenharia, tornando as redes neurais uma ferramenta essencial no desenvolvimento da inteligência artificial moderna.

É importante destacar que, independentemente da sofisticação arquitetural destas redes, seu desempenho está diretamente vinculado à qualidade e quantidade das anotações utilizadas no treinamento. A precisão das anotações de imagens determina o limite superior do desempenho que qualquer arquitetura neural pode alcançar, estabelecendo uma relação direta entre o processo de anotação e o sucesso da rede.

2.2.2 Redes Neurais Convolucionais (RNCs)

As Redes Neurais Convolucionais (RNCs) são utilizadas para análise de imagens e vídeos. Diferentemente das redes neurais tradicionais, as RNCs são projetadas especificamente para processar dados espaciais, aproveitando características como localidade dos pixels e camadas convolucionais para extrair padrões visuais com maior eficiência (LECUN; BENGIO; HINTON, 2015).

A principal vantagem das RNCs reside na sua capacidade de capturar hierarquicamente características visuais, permitindo que camadas iniciais detectem padrões simples, como bordas e texturas, enquanto camadas mais profundas identificam estruturas complexas, como objetos inteiros ou rostos humanos (KRIZHEVSKY SUTSKEVER, 2012).

Uma RNC é composta por diferentes tipos de camadas, cada uma desempenhando um papel específico no processamento das imagens. As principais camadas utilizadas nas RNCs incluem:

1. Camadas Convolucionais: Aplicam filtros (ou kernels) sobre a imagem para extrair características visuais. São responsáveis por aprender padrões e características locais da imagem. Cada filtro detecta padrões específicos, como bordas horizontais, verticais ou curvas. Os pesos desses filtros são ajustados durante o treinamento da rede (LECUN et al., 1998);

2. Camadas de Pooling: Reduzem a dimensão dos mapas de características gerados pela convolução, diminuindo a complexidade computacional e melhorando a generalização do modelo. Os métodos mais comuns são max pooling, que seleciona o valor máximo em uma região, e average pooling, que calcula a média dos valores;

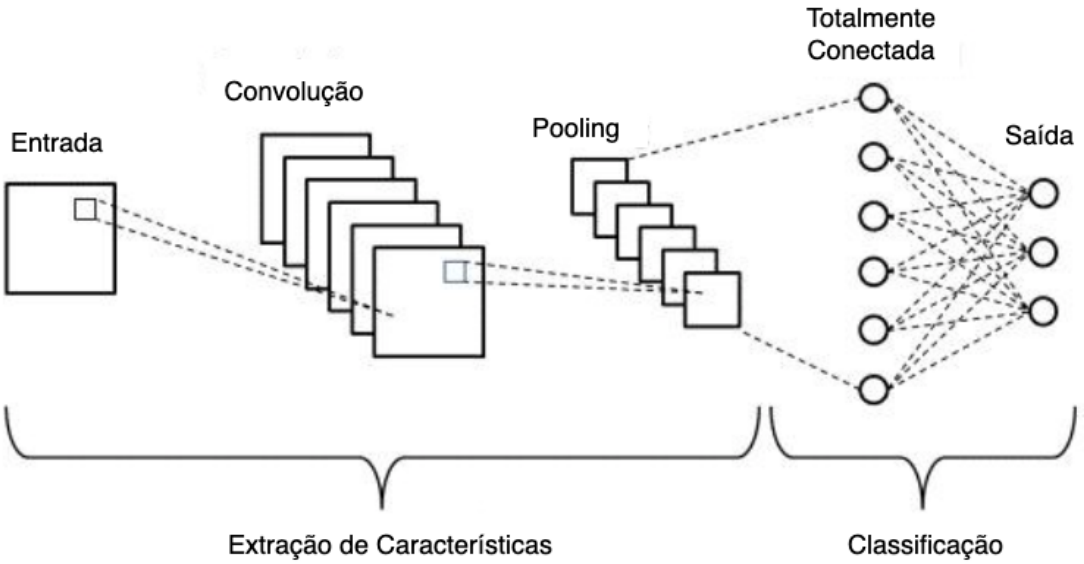
3. Camadas Totalmente Conectadas (Fully Connected Layers): Após a extração de características, as informações são transformadas em um vetor unidimensional e passadas por camadas totalmente conectadas, que realizam a classificação final dos

objetos na imagem;

4. Funções de Ativação: Introduzem não linearidade ao modelo, permitindo que a rede aprenda padrões mais complexos. A função ReLU (Rectified Linear Unit) é amplamente utilizada, pois acelera o treinamento. (GLOROT; BENGIO, 2010).

A Figura 3 ilustra a estrutura geral de uma Rede Neural Convolucional, destacando suas principais camadas e suas funções no processamento de imagens.

Figura 3 – Estrutura Geral de uma RNC



Fonte: Adaptado de ResearchGate (2024).

Para treinar efetivamente estas RNCs, são necessárias anotações específicas que identifiquem precisamente a localização e a categoria dos objetos nas imagens. Diferentes formatos de anotação, como caixas delimitadoras, máscaras de segmentação ou keypoints, permitem que as RNCs aprendam a localizar objetos com precisão crescente. A correspondência entre o tipo de anotação utilizada e a arquitetura da rede é fundamental para maximizar a eficiência do aprendizado.

2.3 Anotação de Imagens para Detecção de Objetos

A anotação de imagens é um processo essencial no treinamento de modelos de visão computacional, sendo responsável por fornecer os rótulos que permitem aos algoritmos aprenderem a identificar e classificar objetos. Diferente de outras áreas do aprendizado de máquina, onde os dados muitas vezes são estruturados e tabulares, os modelos de visão computacional dependem de grandes volumes de imagens anotadas para extração de padrões e generalização eficiente (GOODFELLOW; BENGIO; COURVILLE, 2016).

A qualidade e a precisão das anotações impactam diretamente o desempenho dos modelos, influenciando métricas como acurácia e taxa de falsos positi-

vos/negativos. Uma anotação inconsistente ou incorreta pode comprometer toda a aprendizagem da rede, resultando em modelos que falham em reconhecer corretamente os objetos. Assim, definir uma metodologia eficiente de anotação é um passo crucial no desenvolvimento de sistemas baseados em visão computacional.

Nos últimos anos, diversas técnicas de anotação foram desenvolvidas para atender diferentes aplicações. Métodos como caixas delimitadoras, segmentação semântica, keypoints e máscaras de segmentação são amplamente utilizados em domínios como reconhecimento facial, análise biomédica, monitoramento de tráfego e automação industrial. Além disso, diferentes formatos de armazenamento, como COCO, PASCAL VOC e YOLO, foram criados para estruturar esses dados, garantindo compatibilidade com os frameworks modernos de deep learning.

Esta seção apresenta os principais conceitos relacionados à anotação de imagens, discutindo as técnicas utilizadas, as ferramentas disponíveis e os desafios enfrentados na construção de datasets anotados. A compreensão dessas questões é fundamental para justificar a necessidade da ferramenta proposta neste trabalho, que busca otimizar o processo de anotação para aplicações de visão computacional.

2.3.1 Métodos de Anotação de Imagens

A anotação de imagens define como os dados de treinamento serão apresentados para a rede neural. Dependendo da aplicação e do nível de detalhamento necessário, diferentes métodos de anotação podem ser utilizados.

Modelos modernos de detecção de objetos, como YOLO (You Only Look Once) e Faster R-CNN, são treinados majoritariamente com caixas delimitadoras, enquanto aplicações mais avançadas, como segmentação de imagens médicas, podem exigir métodos mais complexos, como máscaras poligonais e segmentação semântica.

Cada abordagem tem suas vantagens e desvantagens, e a escolha do método de anotação deve considerar fatores como precisão, esforço manual e compatibilidade com o modelo utilizado.

A seguir, serão apresentados os principais métodos de anotação, destacando suas características, aplicações e desafios.

2.3.1.1 Caixas Delimitadoras

A anotação por caixas delimitadoras é a técnica mais comum e amplamente utilizada em tarefas de detecção de objetos. Esse método define a área ocupada pelo objeto dentro de um retângulo, indicando sua posição na imagem. Ela é fundamental para modelos de aprendizado profundo como YOLO (You Only Look Once), Faster R-CNN e SSD (Single Shot MultiBox Detector).

Essas caixas são amplamente utilizadas devido à sua simplicidade, rapidez na anotação e compatibilidade com a maioria dos frameworks de aprendizado de máquina. Contudo, apresentam limitações em casos de objetos com formatos irregulares, onde áreas irrelevantes podem ser incluídas dentro da caixa, reduzindo a precisão do modelo.

2.3.1.2 Definição e Representação

Uma caixa delimitadora é representada por quatro coordenadas que marcam o espaço ocupado pelo objeto na imagem. Os formatos mais comuns de anotação seguem convenções específicas, dependendo da base de dados utilizada. As Figuras 4 e 5 ilustram as representações mais comuns.

a) Coordenadas do canto superior esquerdo (x, y) e largura-altura (w, h).

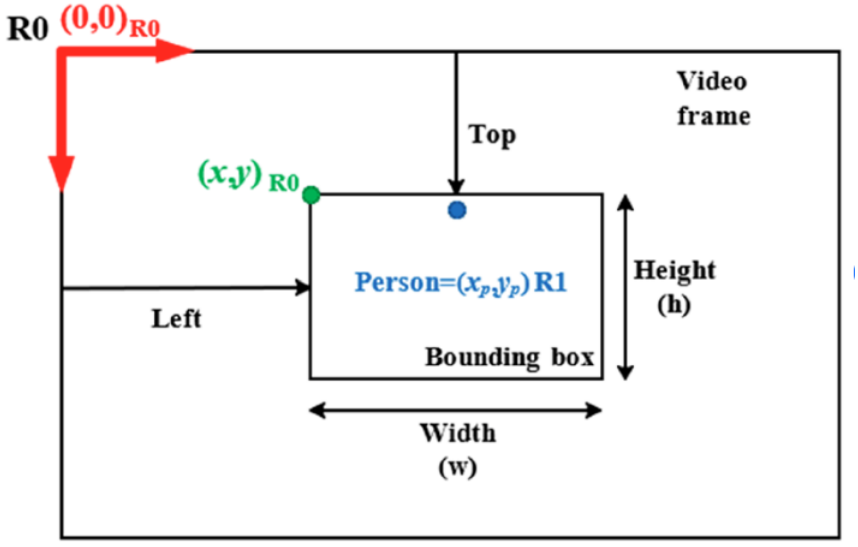
$$B = (x, y, w, h) \tag{1}$$

onde x, y representam as coordenadas do canto superior esquerdo e w, h representam a largura e altura da caixa delimitadora.

b) Coordenadas do centro da caixa x_p, y_p e largura-altura (w, h):

onde x_p e y_p são as coordenadas do centro da caixa, conforme representação 1.

Figura 4 – Caixa Delimitadora pelo Canto Superior Esquerdo e Largura-altura

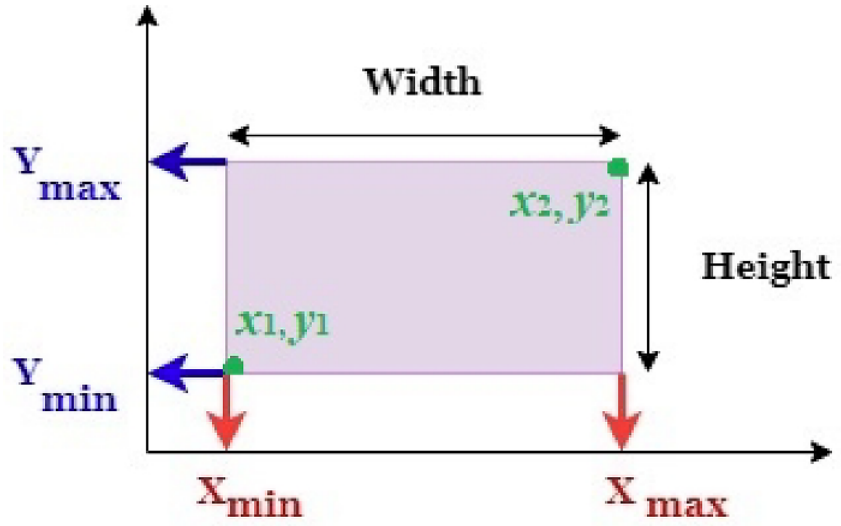


Fonte: Adaptado de ResearchGate (2024).

c) Coordenadas dos quatro vértices x_1, y_1, x_2, y_2.

$$B = (x_1, y_1, x_2, y_2) \tag{2}$$

Figura 5 – Caixa Delimitadora para Cálculos Geométricos Diretos



Fonte: Adaptado de ResearchGate (2024).

Para que redes neurais possam processar corretamente essas caixas, é necessário normalizar os valores das coordenadas. A normalização é feita dividindo os valores pelo tamanho da imagem:

$$x_{\text{norm}} = \frac{x}{W}, \quad y_{\text{norm}} = \frac{y}{H}, \quad w_{\text{norm}} = \frac{w}{W}, \quad h_{\text{norm}} = \frac{h}{H} \tag{3}$$

onde W,H são a largura e altura da imagem, garantindo que as mesmas tenham valores entre 0 e 1, independente da resolução da imagem.

2.3.1.3 Tipos de Caixas Delimitadoras

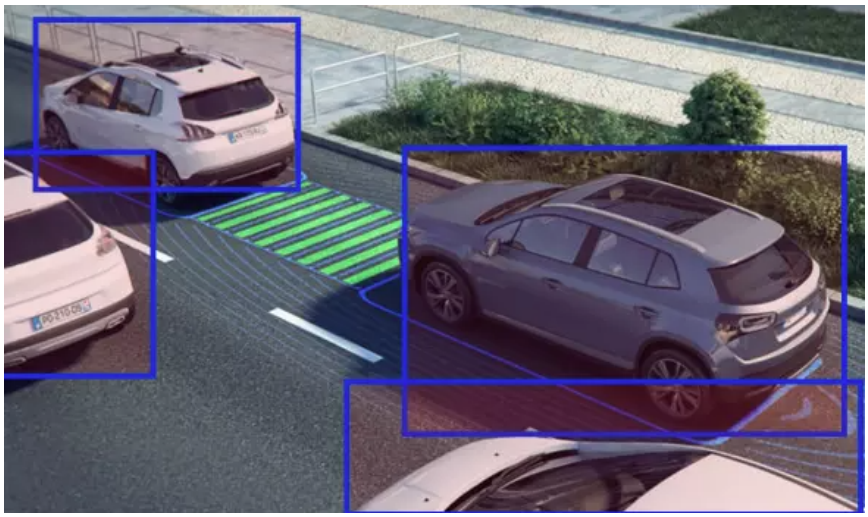
Embora o conceito fundamental de caixas delimitadoras seja relativamente simples, diferentes variações desse método foram desenvolvidas para lidar com desafios específicos encontrados na detecção de objetos em imagens e vídeos. Essas variações buscam melhorar a precisão da delimitação dos objetos, reduzindo áreas ociosas e garantindo maior adaptabilidade em cenários complexos. A seguir, são apresentados os principais tipos utilizados em visão computacional.

Retangulares (Eixo-Alinhadas) - São a forma mais comum de representação de objetos em imagens. Nesse formato, os retângulos seguem a orientação do eixo cartesiano da imagem, ou seja, suas bordas são paralelas aos eixos x e y. Essa abordagem é amplamente utilizada devido à sua simplicidade e eficiência computacional, permitindo rápidas inferências e facilitando a aplicação em modelos de aprendizado profundo.

No entanto, esse método apresenta limitações quando o objeto a ser detectado está inclinado ou possui uma forma irregular. Como a caixa delimitadora retangular

não pode rotacionar para se ajustar ao objeto, muitas vezes ela cobre uma área maior do que a necessária, incluindo regiões que não fazem parte do objeto de interesse. Isso pode comprometer a precisão da detecção, especialmente em domínios onde a orientação do objeto é um fator crítico. A Figura 6 representa uma caixa delimitadora (bounding box) eixo-alinhada.

Figura 6 – Caixa Delimitadora Eixo-Alinhada



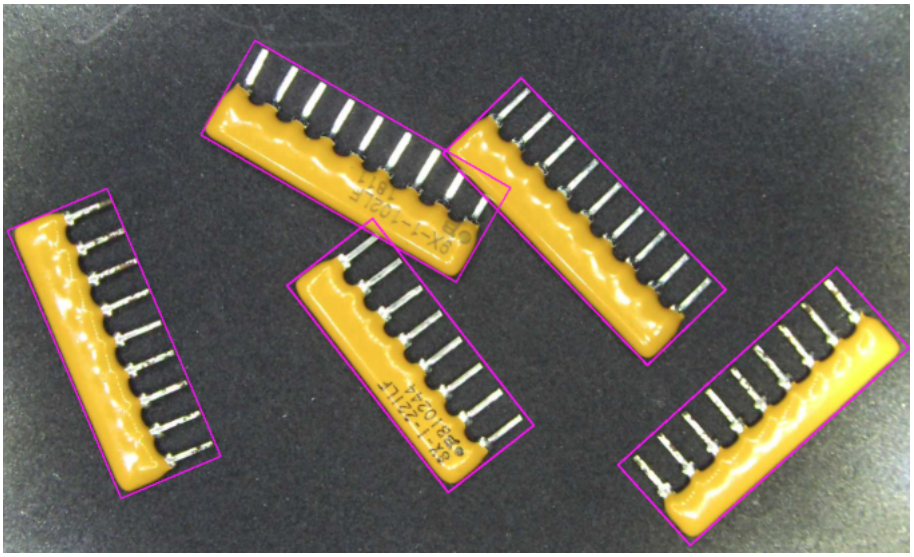
Fonte: Adaptado de Webtunix (2024).

Apesar dessa limitação, são amplamente utilizadas em modelos populares de detecção, como YOLO (You Only Look Once) e Faster R-CNN, devido à sua eficiência na identificação de objetos em tempo real.

Rotacionadas - Para superar as limitações das caixas delimitadoras retangulares, surgiram as rotacionadas, representadas pela Figura 7, também conhecidas como Oriented Bounding Boxes (OBBs). Diferente do modelo tradicional, essa variação permite que o retângulo seja inclinado de acordo com a orientação do objeto, reduzindo a área vazia dentro da caixa. Essa adaptação melhora a precisão da detecção, pois cobre o objeto de maneira mais ajustada, minimizando a inclusão de regiões irrelevantes.

Esse método é especialmente útil em tarefas onde os objetos podem apresentar rotação ou inclinação, como no reconhecimento de texto, onde palavras podem estar dispostas em diferentes ângulos, e na detecção de veículos, especialmente em imagens aéreas, onde os automóveis podem estar orientados de maneira irregular. A aplicação das caixas delimitadoras rotacionadas nesses cenários permite uma representação mais fiel da posição e formato dos objetos, aprimorando a segmentação e reduzindo falsos positivos.

Figura 7 – Caixa Delimitadora Rotacional



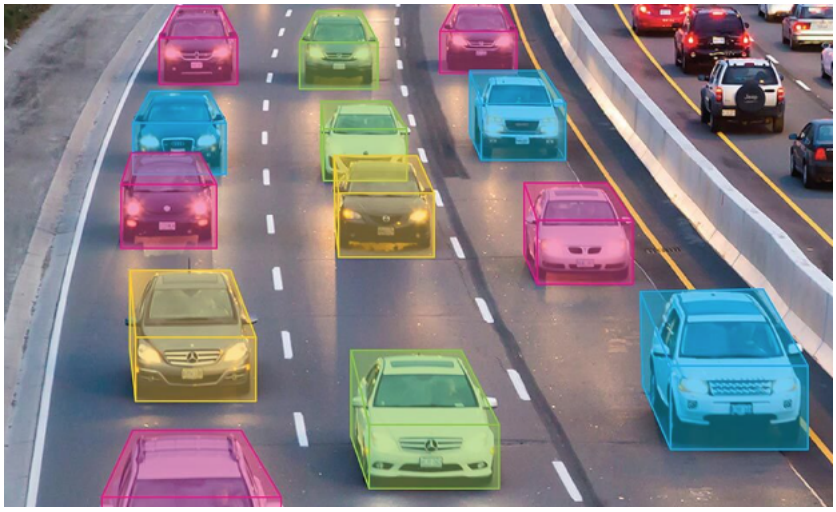
Fonte: Adaptado de Ultralytics (2024).

Embora ofereçam vantagens significativas em termos de precisão, caixas delimitadoras rotacionadas apresentam um custo computacional maior, pois exigem cálculos adicionais para definir o ângulo correto e ajustar as coordenadas do retângulo. Ainda assim, sua aplicação se tornou essencial em domínios que demandam alta precisão na localização dos objetos.

Caixas Delimitadoras 3D - Uma evolução dos modelos tradicionais, se estendem para representar objetos volumétricos no espaço tridimensional. Diferentemente das versões 2D, que apenas delimitam um objeto dentro de uma imagem plana, as caixas delimitadoras 3D incorporam informações de profundidade, permitindo um mapeamento mais detalhado do ambiente.

Esse tipo é amplamente utilizado em tecnologias avançadas, como veículos autônomos, onde a detecção tridimensional de obstáculos é fundamental para a navegação segura. Combinando sensores como LiDAR e câmeras estéreo, os sistemas de visão computacional conseguem construir uma representação 3D do ambiente ao redor do veículo (Figura 8), identificando outros automóveis, pedestres e obstáculos com precisão.

Figura 8 – Caixa Delimitadora 3D



Fonte: Do Autor (2024).

Outra aplicação importante das caixas delimitadoras 3D é na realidade aumentada, onde objetos virtuais precisam ser posicionados corretamente no espaço para interagir com elementos do mundo real. A capacidade de detectar profundidade e volume permite uma melhor integração entre ambientes reais e digitais, tornando a experiência mais imersiva e realista (REDMON *et al.*, 2016).

Apesar das vantagens, as caixas delimitadoras 3D exigem maior poder computacional e sensores mais sofisticados para capturar informações espaciais precisas. Ainda assim, seu uso vem crescendo em diversas áreas, impulsionado pelos avanços na visão computacional e no aprendizado profundo.

As diferentes variações de caixas delimitadoras foram desenvolvidas para atender a necessidades específicas dentro da visão computacional. Enquanto as caixas delimitadoras retangulares continuam sendo a abordagem mais popular devido à sua simplicidade e eficiência, rotacionadas oferecem uma solução mais precisa para objetos inclinados, reduzindo áreas vazias e melhorando a segmentação. Já as 3D representam um avanço significativo, permitindo a detecção e mapeamento de objetos em três dimensões, tornando-se fundamentais para aplicações como veículos autônomos e realidade aumentada.

A escolha do tipo ideal depende do contexto e das exigências da aplicação. Com os avanços contínuos em hardware e algoritmos de aprendizado profundo, é esperado que novas variações e otimizações sejam desenvolvidas, ampliando ainda mais a capacidade de detecção e análise de objetos no campo da visão computacional (LIU *et al.*, 2016).

2.3.2 Outros Métodos de Anotação de Imagens

Embora as caixas delimitadoras sejam amplamente utilizadas por sua simplicidade e eficiência, existem outras técnicas de anotação que oferecem maior precisão e granularidade dependendo da aplicação. Entre elas, destacam-se a segmentação semântica e instanciada e a anotação por keypoints. Cada um desses métodos tem características específicas que os tornam mais adequados para determinados cenários de visão computacional.

2.3.2.1 Segmentação Semântica e Instanciada

A segmentação semântica é uma técnica que atribui um rótulo a cada pixel da imagem, classificando todas as regiões pertencentes a uma determinada classe. Diferentemente das caixas delimitadoras, que fornecem apenas uma área retangular, a segmentação semântica permite que os modelos identifiquem a forma exata dos objetos na imagem. Esse método é amplamente utilizado em diagnóstico médico, onde a identificação precisa de tecidos e órgãos é fundamental, e em veículos autônomos, para diferenciar estradas, pedestres e obstáculos com alta precisão (LONG; SHELHAMER; DARRELL, 2015).

A segmentação instanciada, Figura 9, é uma evolução da segmentação semântica, diferenciando não apenas as classes de objetos, mas também identificando instâncias individuais dentro de uma mesma classe. Enquanto a segmentação semântica pode indicar que há "duas pessoas"na imagem, a segmentação instanciada separa cada uma delas de forma independente. Modelos como Mask R-CNN utilizam essa abordagem para tarefas como reconhecimento facial e análise de cenas complexas.

Figura 9 – Exemplo de segmentação semântica e instanciada em uma cena urbana

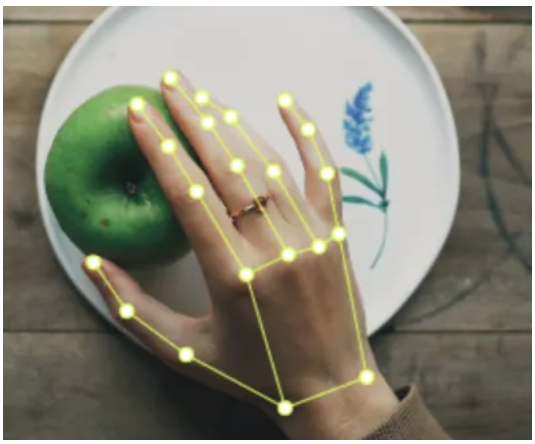
Fonte: Adaptado de Medium (2017).

2.3.2.2 Anotação por Keypoints

A anotação por keypoints (pontos-chave) é um método que marca pontos específicos em um objeto para capturar informações sobre sua forma e estrutura. Essa técnica é amplamente utilizada para detecção de poses humanas, reconhecimento facial e rastreamento de movimentos. Em aplicações como animação 3D, análise de esportes e monitoramento de atividades, a anotação por keypoints permite identificar articulações e movimentos com alta precisão (CAO *et al.*, 2017).

Modelos como OpenPose utilizam esse método para mapear articulações humanas e prever posturas em tempo real. Além disso, sistemas de reconhecimento facial modernos aplicam keypoints para identificar características faciais, como olhos, nariz e boca, possibilitando o desenvolvimento de tecnologias de autenticação biométrica, como mostrado na Figura 10.

Figura 10 – Exemplo de anotação por keypoints para detecção de mão humana



Fonte: V7Labs (2017).

A escolha do método de anotação tem um impacto direto na qualidade do modelo de visão computacional. Caixas delimitadoras continuam sendo uma das abordagens mais utilizadas devido à sua eficiência e simplicidade, sendo a técnica central adotada na ferramenta desenvolvida neste trabalho. No entanto, métodos como segmentação semântica e keypoints oferecem soluções mais detalhadas para problemas que exigem maior precisão. A seguir, será apresentada uma análise das ferramentas disponíveis para anotação de imagens, destacando aquelas que permitem a geração de datasets de alta qualidade para treinar modelos de detecção de objetos.

2.3.3 Ferramentas de Anotação de Imagens

O LabelImg e o Roboflow Annotate são duas das ferramentas amplamente utilizadas para a anotação de imagens no contexto da detecção de objetos e serviram como referência para o estudo e desenvolvimento do trabalho. O LabelImg, uma ferramenta open-source, permite a marcação manual de objetos por meio de caixas

delimitadoras e é amplamente adotado por projetos que utilizam YOLO e PASCAL VOC como formatos de dataset. Já o Roboflow Annotate combina anotação manual e inteligência artificial para otimizar o processo de rotulagem, oferecendo suporte a múltiplos formatos e integração direta com modelos de aprendizado profundo. Ambos desempenham um papel crucial na construção de datasets para visão computacional, e a escolha entre eles depende da escala do projeto e da necessidade de automação no processo de anotação.

2.3.3.1 LabelImg

O LabelImg é uma ferramenta de anotação de imagens leve e eficiente, desenvolvida em Python com a biblioteca PyQt. Seu foco principal é a criação de caixas delimitadoras, permitindo que os usuários rotulem rapidamente imagens para treinar modelos de detecção de objetos. Ele é frequentemente utilizado em projetos que exigem datasets compatíveis com YOLO, PASCAL VOC e TensorFlow Object Detection API. A Figura 11 representa a interface de usuário do LabelImg.

Figura 11 – Interface do LabelImg com anotações em caixas delimitadoras

Fonte: Tzutalin (2018).

Características do LabelImg: Interface simples e intuitiva, permitindo anotações rápidas. Compatível com YOLO, PASCAL VOC e COCO, facilitando a exportação para diversos frameworks. Atalhos de teclado para marcação ágil, reduzindo o tempo de anotação manual. Execução offline, sem necessidade de conexão com a internet. A principal vantagem do LabelImg está na sua simplicidade e eficiência. A ferramenta permite que usuários realizem anotações de forma precisa e estruturada, garantindo que as caixas delimitadoras estejam bem ajustadas aos objetos de interesse. No en-

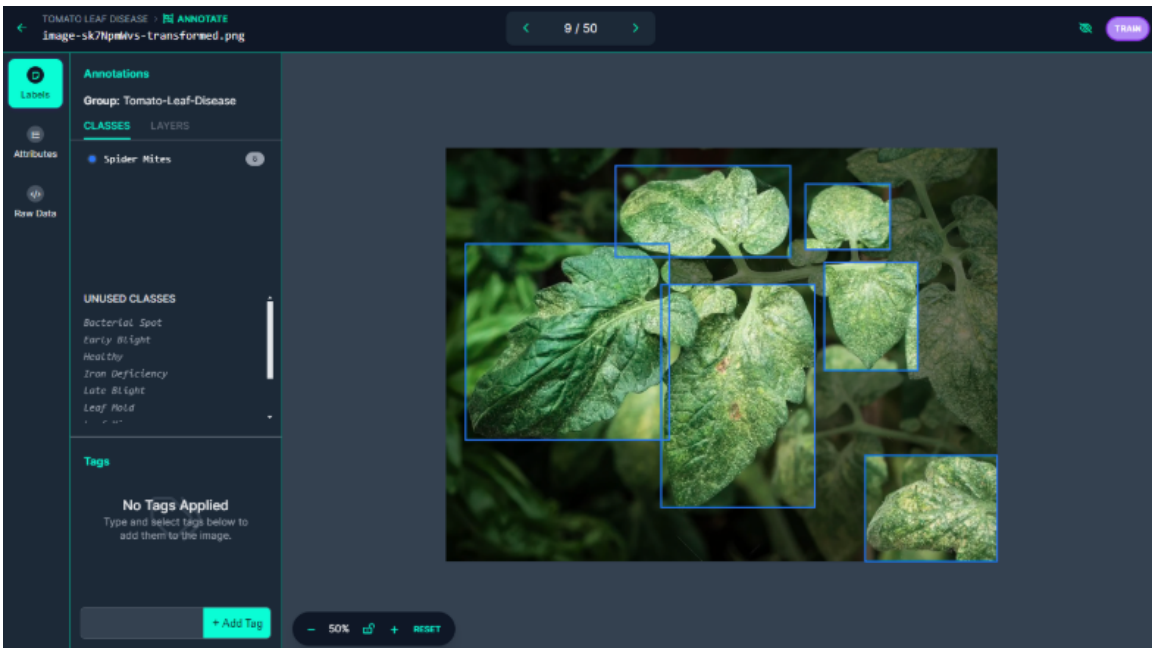
tanto, por ser totalmente manual, torna-se inviável para projetos de larga escala, onde a anotação manual pode se tornar um gargalo (TZUTALIN, 2015).

No escopo deste trabalho, a ferramenta desenvolvida busca oferecer uma solução inicial eficiente e escalável para anotação de imagens, garantindo compatibilidade com formatos amplamente utilizados, como YOLO. Embora o LabelImg tenha sido uma ferramenta consolidada nesse contexto, seu desenvolvimento foi descontinuado, o que pode comprometer sua manutenção e atualização para atender às necessidades atuais. Diante disso, a ferramenta proposta se destaca por fornecer um ambiente atualizado e funcional para anotação de imagens, assegurando a expansão do projeto para o treinamento de modelos de aprendizado profundo.

2.3.3.2 Roboflow Annotate

O Roboflow Annotate é uma plataforma moderna paga baseada em nuvem, disponibilizada por meio de um modelo pago, que une anotação manual e inteligência artificial para tornar o processo de marcação de imagens mais eficiente. Diferente do LabelImg, que exige que cada objeto seja manualmente rotulado, o Roboflow permite a pré-anotação automática, onde modelos já treinados fazem sugestões que o usuário pode corrigir e validar. Essa abordagem reduz significativamente o tempo necessário para preparar datasets para treinamento. A Figura 12 mostra a interface de anotação do Roboflow.

Figura 12 – Interface do Roboflow Annotate



Fonte: Roboflow (2022).

Características do Roboflow Annotate: Automação no processo de anotação, reduzindo o esforço manual. Anotação colaborativa em tempo real, permitindo que

equipes trabalhem simultaneamente. Suporte para diferentes formatos de dataset, incluindo YOLO, COCO e PASCAL VOC. Conversão automática de formatos, facilitando a adaptação entre diferentes frameworks. O grande diferencial do Roboflow está na sua capacidade de integração com modelos de aprendizado profundo, permitindo a anotação assistida por IA. Essa funcionalidade reduz inconsistências humanas, garantindo maior padronização nas anotações. No entanto, sua dependência da nuvem e planos pagos para algumas funcionalidades avançadas podem ser limitantes em alguns contextos (ROBOFLOW, 2021).

A seguir, serão discutidos os desafios enfrentados na construção de datasets, bem como estratégias para garantir um processo de anotação eficiente e preciso.

2.3.4 Construção de Datasets Anotados

A organização e estruturação dos datasets anotados desempenham um papel essencial na implementação de modelos de visão computacional. Durante a anotação de imagens, além da marcação dos objetos de interesse, é necessário que os dados sejam salvos em formatos específicos compatíveis com os frameworks de treinamento. Esses formatos definem a forma como as informações de anotação são armazenadas, incluindo detalhes como coordenadas das caixas delimitadoras, classes dos objetos e metadados da imagem. Diferentes métodos de anotação adotam estruturas de arquivos distintas, cada uma com vantagens e desvantagens dependendo da aplicação.

Os formatos de dataset mais comuns utilizados na anotação de imagens incluem YOLO, PASCAL VOC e COCO, cada um apresentando um esquema de armazenamento específico. Embora este trabalho utilize exclusivamente o formato YOLO, é essencial compreender a estrutura dos demais formatos para contextualizar a construção dos datasets anotados.

2.3.4.1 YOLO

O formato YOLO (*You Only Look Once*) é amplamente utilizado em modelos de detecção de objetos devido à sua estrutura simples e eficiente. Cada imagem anotada no formato YOLO possui um arquivo de texto (`.txt`) associado, onde são armazenadas as coordenadas das caixas delimitadoras juntamente com a classe do objeto. A estrutura de um arquivo YOLO segue o seguinte padrão, conforme ilustrado na Figura 13.

A estrutura de um arquivo YOLO segue o seguinte padrão:

Figura 13 – Formato de saída arquivo YOLO

$$\text{<class>} \; \text{<}x_c\text{>} \; \text{<}y_c\text{>} \; \text{<}width\text{>} \; \text{<}height\text{>}$$

Fonte: Do Autor (2024).

Onde:

$class_id$: Identificação da classe do objeto (um número inteiro associado a uma classe no arquivo de classes).

x_c, y_c: Coordenadas do centro da caixa delimitadora (normalizadas entre 0 e 1). width, height: Largura e altura da caixa delimitadora (também normalizadas entre 0 e 1).

Juntamente com o arquivo `.txt`, a imagem original (`.jpg`, `.png`) deve ser mantido no mesmo diretório para ser utilizado no treinamento.

2.3.4.2 PASCAL VOC

O formato PASCAL VOC é estruturado em arquivos XML, que armazenam informações detalhadas sobre a anotação da imagem, incluindo coordenadas da caixa delimitadora, nome da classe, resolução da imagem e outros metadados. A estrutura de um arquivo PACAL VOC segue o seguinte padrão, ilustrado na Figura 14.

Exemplo de estrutura XML para uma anotação no formato PASCAL VOC:

Figura 14 – Formato de saída arquivo Pascal VOC

```
<annotation>
    <folder>dataset</folder>
    <filename>image1.jpg</filename>
    <size>
        <width>640</width>
        <height>480</height>
        <depth>3</depth>
    </size>
    <object>
        <name>cachorro</name>
        <bndbox>
            <xmin>100</xmin>
            <ymin>200</ymin>
            <xmax>400</xmax>
            <ymax>500</ymax>
        </bndbox>
    </object>
</annotation>
```

Fonte: Do Autor (2024).

Esse formato é utilizado por frameworks como TensorFlow Object Detection

API, sendo mais detalhado do que o YOLO, mas também mais pesado, exigindo maior esforço computacional para processamento.

2.3.4.3 COCO

O formato COCO (Common Objects in Context) é mais avançado e adequado para tarefas de detecção de objetos, segmentação e keypoints. Diferente do YOLO e do PASCAL VOC, o COCO utiliza um único arquivo JSON para armazenar todas as anotações do dataset, incluindo informações detalhadas sobre cada imagem, como categorias, caixas delimitadoras e segmentação de polígonos, seu formato é ilustrado na Figura 15.

Exemplo de anotação no formato COCO:

Figura 15 – Formato de saída arquivo COCO, formato JSON

```
{
    "annotations": [
        {
            "image_id": 1,
            "category_id": 1,
            "bbox": [100, 200, 300, 400],
            "segmentation": [[120, 220, 150, 250, 300, 400]]
        }
    ]
}
```

Fonte: Do Autor (2024).

O COCO é um dos formatos mais estruturados e versáteis, sendo utilizado em grandes competições de visão computacional, mas sua complexidade torna-o mais difícil de manipular para aplicações mais simples.

2.4 Trabalhos Correlatos

O desenvolvimento de ferramentas para anotação de imagens desempenha um papel central na construção de datasets para visão computacional. Nos últimos anos, diversas plataformas e estudos acadêmicos abordaram soluções para melhorar o processo de anotação, garantindo maior precisão e eficiência no treinamento de modelos de aprendizado profundo. Nesta seção, serão discutidos alguns trabalhos relevantes que propõem soluções para a anotação de imagens, focando principalmente em métodos baseados em caixa delimitadora e comparando essas abordagens com a ferramenta desenvolvida neste trabalho.

2.4.1 Ferramentas de Anotação Existentes

A criação de datasets anotados envolve o uso de ferramentas que permitem ao usuário marcar regiões de interesse dentro das imagens e exportar essas anotações em formatos padronizados. Entre as principais ferramentas já estabelecidas na literatura e no mercado, destacam-se:

LabelImg: Um dos softwares mais utilizados para anotação manual de caixa delimitadora. Desenvolvido como uma aplicação Python utilizando a biblioteca Qt, o LabelImg permite ao usuário desenhar caixas delimitadoras sobre objetos em imagens e salvar as anotações no formato PASCAL VOC (`.xml`) ou YOLO (`.txt`). No entanto, a ferramenta apresenta limitações quanto à usabilidade, sendo necessário que o usuário ajuste manualmente a posição das caixas, além de não oferecer funcionalidades avançadas como segmentação ou automação na anotação (TZUTALIN, 2015).

Roboflow: Plataforma online que oferece um ambiente completo para anotação, pré-processamento e treinamento de modelos de visão computacional. Diferente de ferramentas locais, o Roboflow permite a conversão entre diferentes formatos de anotação, além de incluir recursos como aumento de dados (*data augmentation*) e integração com frameworks populares. No entanto, para anotações offline e manipulação direta dos arquivos, sua dependência da nuvem pode ser um fator limitante para projetos que necessitam de maior controle sobre o ambiente de anotação (ROBOFLOW, 2021).

CVAT (Computer Vision Annotation Tool): Desenvolvida pela Intel, essa ferramenta foi projetada para lidar com grandes volumes de imagens e oferece suporte para múltiplos métodos de anotação, incluindo caixas delimitadoras, segmentação e keypoints. Seu diferencial está na capacidade de anotação colaborativa e automação parcial por meio de modelos pré-treinados, facilitando a rotulagem em larga escala. No entanto, sua complexidade de uso e dependência de servidores tornam sua aplicação menos prática para anotações locais simples (CORPORATION, 2019).

2.4.2 Diferenças e Contribuições da Ferramenta Desenvolvida

As ferramentas existentes oferecem soluções variadas para anotação de imagens, mas possuem limitações que motivam o desenvolvimento de alternativas mais especializadas. A ferramenta proposta neste trabalho diferencia-se por focar exclusivamente na anotação de caixas delimitadoras no formato YOLO, proporcionando um fluxo de trabalho otimizado e direto para aplicações que utilizam esse formato.

Enquanto plataformas como Roboflow e CVAT oferecem funcionalidades avançadas de anotação e integração com frameworks de deep learning, estas características frequentemente vêm acompanhadas de maior complexidade de uso, curvas de aprendizado mais acentuadas e, em muitos casos, restrições como custos associados

a recursos premium. A ferramenta desenvolvida neste estudo, por outro lado, adota uma abordagem complementar que prioriza a simplicidade, modularidade e eficiência.

O objetivo principal é fornecer uma solução web que permita a pesquisadores, estudantes e profissionais realizarem anotações de forma rápida e intuitiva, sem a necessidade de configurações complexas ou dependência de plataformas proprietárias. Esta abordagem baseada na web facilita a compatibilidade com praticamente qualquer dispositivo com navegador moderno, independente do sistema operacional ou configuração de hardware. Diferentemente de aplicações desktop que podem requerer bibliotecas específicas ou dependências de sistema, a solução web elimina barreiras de compatibilidade e permite acesso imediato sem necessidade de configurações adicionais.

Além disso, a ferramenta foi projetada para ser escalável, permitindo que futuras melhorias sejam incorporadas de maneira modular. Atualmente, a funcionalidade se concentra na anotação de caixas delimitadoras para o formato YOLO, mas sua estrutura pode ser adaptada para incluir novos métodos de anotação, como segmentação semântica e keypoints, ou até mesmo incorporar algoritmos de deep learning para pré-anotação automática e sugestão de objetos. Dessa forma, o projeto serve como base para trabalhos futuros, seja adicionando suporte a outros formatos de anotação ou integrando funcionalidades de automação na rotulagem de imagens.

3 METODOLOGIA

Neste capítulo, é apresentada a fase de desenvolvimento da ferramenta de anotação de imagens. O objetivo é descrever de forma detalhada os processos e metodologias adotados na criação desta aplicação, destacando as escolhas técnicas, os desafios enfrentados e as soluções implementadas. O desenvolvimento de uma ferramenta como esta, destinada a facilitar a anotação de imagens em ambientes baseados na web, não é apenas uma contribuição técnica para o campo da visão computacional, mas também uma resposta prática às necessidades de anotação de imagens de forma mais acessível e eficiente.

A estrutura deste capítulo segue uma abordagem sistemática, iniciando pela definição dos requisitos funcionais e não funcionais. Esses requisitos estabelecem a base sobre a qual a ferramenta foi projetada e desenvolvida, delimitando claramente seu escopo e objetivos. Em seguida, é descrito o projeto da solução proposta, detalhando sua arquitetura e apresentando diagramas que ilustram o fluxo de trabalho e a estrutura da aplicação. Esses elementos são essenciais para compreender como a ferramenta foi concebida para atender às necessidades identificadas.

O núcleo deste capítulo está na seção de desenvolvimento, onde é abordado o processo de construção da aplicação, desde a seleção das tecnologias até a implementação prática do código. Também são discutidos os desafios técnicos encontrados ao longo do desenvolvimento e as estratégias adotadas para superá-los, proporcionando uma visão transparente do processo.

A análise dos resultados é apresentada em um capítulo específico, onde são demonstrados casos de uso que validam a aplicação prática da ferramenta e sua eficácia. Essa avaliação inclui exemplos de imagens anotadas e os arquivos gerados, demonstrando a funcionalidade e a utilidade da ferramenta desenvolvida.

A conclusão deste capítulo resume o impacto do projeto, avaliando como a solução atendeu aos requisitos e objetivos estabelecidos. Além disso, são discutidas as lições aprendidas e possíveis melhorias que poderão orientar desenvolvimentos futuros na área.

3.1 Definição dos Requisitos

A ferramenta foi projetada para ser intuitiva e focada exclusivamente na anotação de bounding boxes no formato YOLO. Os requisitos foram divididos em funcionais e não funcionais, garantindo que o escopo do projeto estivesse bem definido.

A aplicação desenvolvida busca otimizar o processo de anotação manual de imagens para modelos de visão computacional, especificamente para caixas delimitadoras. Diferente de ferramentas existentes, como LabelImg e Roboflow Annotate, esta

solução é baseada na web, oferecendo maior acessibilidade e escalabilidade sem a necessidade de instalação local ou dependência de plataformas pagas.

O LabelImg, por ser offline e executado localmente, é útil para projetos menores, mas se torna inviável para equipes ou para grandes volumes de dados devido à falta de recursos colaborativos e otimizações. Além disso, seu desenvolvimento foi descontinuado, o que pode impactar sua manutenção e compatibilidade com novas demandas.

O Roboflow, por outro lado, facilita a anotação em nuvem e integra automação parcial, mas impõe limitações no acesso gratuito, exige conexão constante com a internet e não é totalmente otimizado para fluxos puramente manuais.

A proposta deste trabalho combina a flexibilidade de uma aplicação web com a autonomia da anotação manual, permitindo que usuários realizem rótulos diretamente no navegador, garantindo que os dados sejam centralizados e facilmente acessíveis. Além disso, a arquitetura da ferramenta permite possíveis melhorias futuras, como integração com modelos de sugestão de caixas delimitadoras, exportação facilitada para múltiplos formatos e suporte para trabalho colaborativo.

Ao unir acessibilidade, eficiência e escalabilidade, essa ferramenta se posiciona como uma alternativa mais equilibrada para projetos que necessitam de anotação manual sem comprometer a usabilidade e a qualidade do dataset.

3.1.1 Requisitos Funcionais

1. Carregamento de Imagens: A capacidade de carregar imagens no formato .jpg, .png e outros formatos comuns diretamente na interface da ferramenta.

2. Criação de Caixas Delimitadoras: Ferramenta intuitiva para desenhar caixas delimitadoras retangulares eixo-alinhadas ao redor dos objetos nas imagens.

3. Atribuição de Etiquetas: A possibilidade de atribuir etiquetas de classe a cada bounding box, facilitando a categorização dos objetos anotados.

4. Salvar/Deletar e Exportar Anotações: Funcionalidade para salvar, ou excluir as anotações realizadas, e exportá-las em formatos compatíveis com ferramentas de aprendizado de máquina, como o formato YOLO.

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais estão relacionados às características gerais da aplicação, como desempenho, segurança e usabilidade. Estes incluem:

1. Interface Simplificada: A interface do usuário é direta e objetiva, abrindo diretamente na tela de anotação sem necessidade de etapas adicionais como login.

2. Execução Local e Potencial de Hospedagem: Embora atualmente a ferramenta seja configurada para execução local (localhost), o código foi desenvolvido de forma a permitir uma futura hospedagem em servidores, para ampliar seu acesso, utilidade e aprimoramento.

3. Compatibilidade com Navegadores: Projetada para funcionar eficientemente em navegadores web modernos, garantindo boa performance mesmo sem recursos de responsividade.

Esses requisitos foram estabelecidos para garantir que a ferramenta não apenas atendesse às necessidades imediatas de anotação de imagens, mas também fosse adaptável para expansões e melhorias futuras, alinhando-se com os objetivos de longo prazo do projeto.

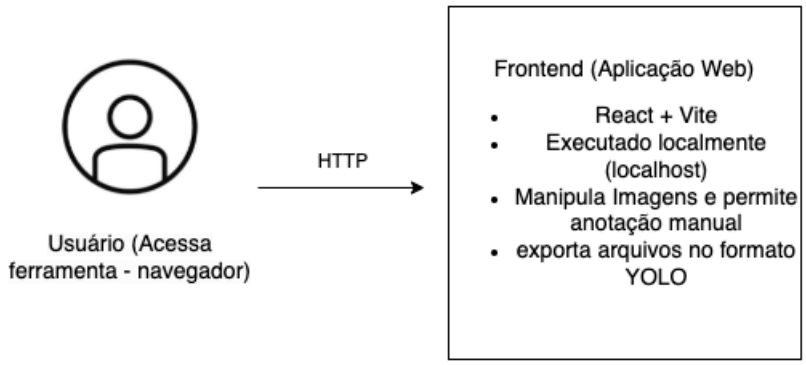
3.2 Projeto da Solução Proposta

Nesta seção, é descrito o projeto da ferramenta de anotação de imagens baseada na web, detalhando sua arquitetura e design funcional. A intenção é apresentar uma visão clara de como os componentes da aplicação interagem entre si e como a ferramenta foi projetada para atender aos requisitos definidos.

3.2.1 Arquitetura do Sistema

A arquitetura desta ferramenta de anotação de imagens, ilustrada pelo diagrama de componentes da figura 16, foi planejada para assegurar eficiência e usabilidade. Construída inteiramente como uma aplicação de frontend, foi desenvolvida utilizando JavaScript, com o framework React Vite, otimizado para performance e escalabilidade. A escolha do React Vite está alinhada com a necessidade de uma aplicação que utilize conceitos modernos e de alta performance, capaz de manipular a interatividade do usuário e a atualização de dados de maneira eficiente.

Figura 16 – Diagrama de Contexto, Anotador de Imagem.



Fonte: Do Autor (2024).

A biblioteca react-image-annotation é integrada como um componente chave, fornecendo a funcionalidade essencial de anotação de imagens. Esta biblioteca foi escolhida por sua flexibilidade e capacidade de ser customizada para atender às necessidades específicas do projeto. Ela permite que os usuários desenhem bounding boxes diretamente sobre as imagens carregadas e associem etiquetas a estas anotações de forma intuitiva.

3.2.2 Tecnologias Utilizadas

Para garantir eficiência, modularidade e escalabilidade, a ferramenta foi construída utilizando React.js no frontend, juntamente com Tailwind CSS para estilização e diversas bibliotecas especializadas para manipulação de imagens e arquivos.

3.2.2.1 JavaScript e React.js

A escolha do React.js como framework principal foi motivada por diversos fatores:

1. Gerenciamento eficiente de estado: O uso de React Hooks (useState, useEffect) permite atualização dinâmica da interface sem necessidade de recarregar a página. 2. Componentização: O código foi estruturado em componentes reutilizáveis, garantindo modularidade e manutenção facilitada. 3. Virtual DOM: React otimiza a renderização de componentes, garantindo melhor desempenho. 4. Facilidade de expansão: O framework permite adição de novas funcionalidades sem reescrever grandes partes do código.

A aplicação foi estruturada seguindo o padrão SPA (Single Page Application), garantindo interação fluida sem recarregamentos desnecessários.

3.2.2.2 Tailwind CSS

Para estilização da interface, foi utilizado o Tailwind CSS, um framework CSS que oferece:

1. Classes utilitárias: Permite estilização rápida sem necessidade de criar arquivos CSS extensos. 2. Flexibilidade: Personalização simplificada para atender às necessidades do projeto. 3. Desempenho otimizado: Remove estilos não utilizados em produção, garantindo arquivos CSS leves.

O uso do Tailwind permitiu a construção de uma interface simples e funcional, com foco na usabilidade.

3.2.2.3 Bibliotecas Adicionais

Além do React e Tailwind, foram utilizadas outras bibliotecas para funções específicas:

1. React Image Annotation: Responsável por gerenciar a anotação de imagens dentro da aplicação. 2. File System Access API: Permite leitura e gravação de arquivos diretamente no navegador. 3. React Icons: Fornece ícones vetoriais para melhorar a experiência do usuário.

A escolha dessas bibliotecas garantiu que a ferramenta fosse leve e sem dependências desnecessárias, focando na funcionalidade principal.

3.2.3 Considerações de Design

O design da interface do usuário foi concebida com o objetivo de maximizar a eficiência e minimizar a complexidade. A interface (figura 17) é limpa e sem elementos desnecessários, permitindo que os usuários se concentrem na tarefa de anotar imagens. Foi dada atenção à facilidade de uso, garantindo que mesmo usuários sem experiência prévia em ferramentas de anotação possam utilizar a aplicação com pouco ou nenhum treinamento.

Figura 17 – Interface Web, Anotador de Imagem.

Fonte: Do Autor (2024).

Além disso, o código foi estruturado de maneira a facilitar a manutenção e a extensibilidade. Isso permite que outros desenvolvedores e pesquisadores adaptem e expandam a aplicação no futuro, seja para adicionar novas funcionalidades ou para integrá-la em sistemas mais amplos de processamento de imagens.

3.2.4 Estrutura do Projeto

A estrutura do projeto foi organizada para garantir clareza, modularidade e facilidade de manutenção. O código-fonte foi separado em diretórios bem definidos:

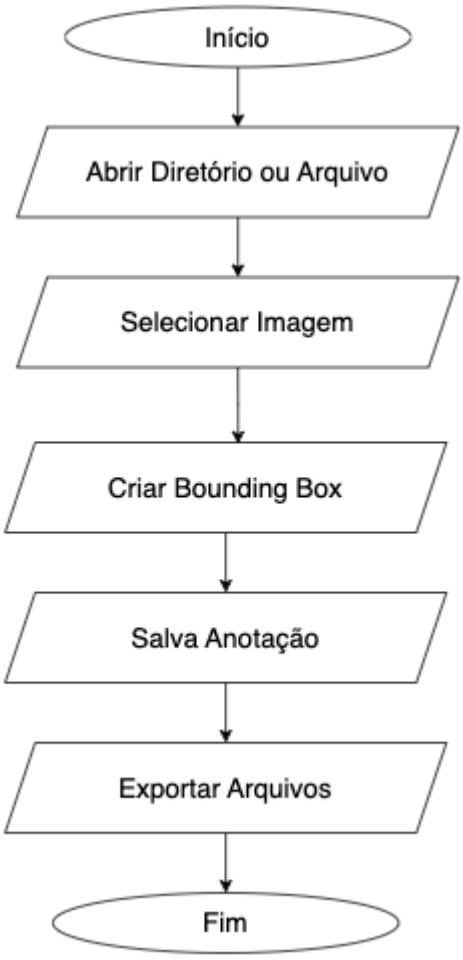
```
/labelimg-web
     src/
        assets/              # Recursos visuais
        components/          # Componentes reutilizáveis
           ImageArea.jsx     # Área de exibição da imagem e anotações
           Sidebar.jsx       # Barra lateral para controle de arquivos
           Toolbar.jsx       # Ferramentas para manipulação das bounding boxes
        App.jsx              # Componente principal da aplicação
        index.js             # Ponto de entrada do React
     public/                 # Arquivos estáticos
     package.json            # Configuração das dependências do projeto
     vite.config.js          # Configuração do Vite para otimização do build
```

Essa estrutura modular permite expansões futuras, facilitando a adição de novos métodos de anotação ou suporte a outros formatos além do YOLO.

3.2.5 Diagramas de Usuário e Componetes

Fluxo de usuário: Este diagrama ilustra o percurso do usuário dentro da aplicação, desde o carregamento inicial da página até a finalização das anotações. O fluxo detalha cada etapa da interação do usuário, incluindo o carregamento de imagens, a criação e edição de bounding boxes, e a exportação das anotações.

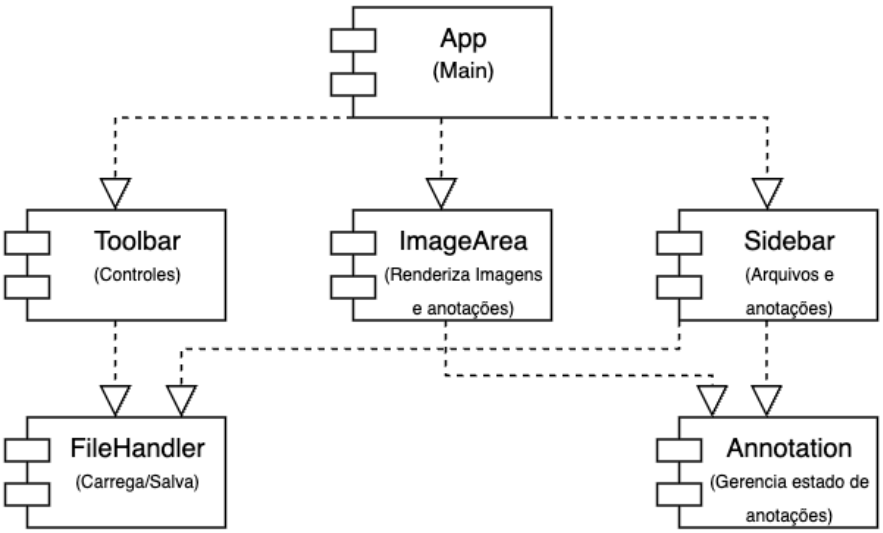
Figura 18 – Diagrama de Usuário, Anotador de Imagem.



Fonte: Do Autor (2024).

Diagrama de Componentes: Este diagrama desdobra a estrutura da aplicação, mostrando como os diferentes componentes do React interagem entre si. O foco é demonstrar a modularidade da aplicação, com componentes distintos para carregamento de imagens, interface de anotação e salvamento de dados.

Figura 19 – Diagrama de Componentes, Anotador de Imagem.



Fonte: Do Autor (2024).

A aplicação é estruturada modularmente para facilitar a separação de responsabilidades. A seguir, explico os principais componentes e suas funções:

3.2.5.1 Componentes da Aplicação

a) App (Componente Principal)

O componente App é o núcleo da aplicação. Ele gerencia todo o estado global do sistema e é responsável por conectar os diferentes componentes, garantindo que a navegação entre imagens e a manipulação das anotações ocorra de forma fluida e eficiente.

a) Armazena e gerencia a lista de arquivos carregados pelo usuário na variável fileList, que contém as imagens que serão anotadas.
b) Mantém o índice da imagem atual por meio da variável currentIndex, permitindo que o usuário navegue entre os arquivos carregados.
c) Controla o estado das anotações de cada imagem, garantindo que cada anotação seja corretamente associada ao respectivo arquivo de imagem.
d) Implementa as funções nextImage e prevImage para permitir que o usuário avance ou volte para a próxima imagem na lista.
e) Interage diretamente com os componentes Toolbar, Sidebar e ImageArea, garantindo que cada um receba os dados necessários para operar corretamente.

b) Toolbar (Barra de Ferramentas)

A barra de ferramentas Toolbar é responsável por oferecer ao usuário opções para carregar imagens, navegar entre os arquivos e salvar as anotações no formato desejado.

a) Permite que o usuário abra um único arquivo de imagem ou selecione uma pasta contendo várias imagens.

b) Exibe botões para navegação entre as imagens carregadas, permitindo avançar (Next Image) ou retroceder (Previous Image).

c) Inclui um botão específico para salvar as anotações no formato YOLO.

d) Contém uma interface para interação com o sistema de arquivos, garantindo que os arquivos de imagem e anotações possam ser carregados e salvos corretamente.

c) Sidebar (Painel Lateral)

O painel lateral Sidebar fornece uma interface para visualizar e gerenciar as anotações feitas em cada imagem, além de exibir a lista de arquivos carregados.

a) Exibe todas as anotações criadas pelo usuário para a imagem atualmente selecionada.

b) Permite que o usuário exclua anotações individuais diretamente no painel.

c) Atualiza dinamicamente a interface conforme o usuário adiciona ou remove anotações.

d) Implementa rolagem vertical para que uma grande quantidade de anotações possa ser visualizada sem comprometer o layout da aplicação.

e) Mostra a lista de imagens carregadas e destaca a imagem atualmente selecionada.

d) ImageArea (Área de Visualização e Anotação)

A área de visualização e anotação ImageArea é responsável por renderizar a imagem carregada e fornecer ferramentas para que o usuário desenhe bounding boxes e crie anotações.

a) Renderiza a imagem carregada pelo usuário na tela, garantindo que ela seja corretamente ajustada ao espaço disponível.

b) Permite que o usuário desenhe bounding boxes ao redor dos objetos de interesse na imagem.

c) Atualiza as anotações em tempo real conforme o usuário interage com a ferramenta.

d) Garante que as bounding boxes respeitem os limites da imagem carregada.

e) Interage diretamente com o estado global da aplicação, garantindo que todas as anotações sejam salvas corretamente.

e) Annotation System (Sistema de Anotações)

O sistema de anotações Annotation System processa os dados gerados pelo usuário e os converte para o formato YOLO, garantindo que possam ser utilizados para treinamento de modelos de aprendizado de máquina.

a) Coleta as anotações feitas pelo usuário e converte as coordenadas das bounding boxes para o formato normalizado do YOLO.

b) Gera um arquivo de texto para cada imagem anotada, contendo as informações das bounding boxes e suas respectivas classes.

c) Mantém um mapa de classes (labelmap.txt) que relaciona cada categoria de objeto com um identificador numérico, garantindo que todas as anotações sigam um padrão consistente.

f) File System (Manipulação de Arquivos)

O sistema de manipulação de arquivos File System gerencia a leitura e escrita dos arquivos de imagem e anotações no sistema do usuário.

a) Permite a seleção de um diretório para salvar automaticamente todas as anotações em uma pasta específica (annotations_yolo).

b) Gerencia a leitura de imagens do diretório escolhido, garantindo que apenas arquivos de imagem sejam carregados na aplicação.

c) Manipula a escrita dos arquivos de anotação no formato YOLO, garantindo que cada anotação seja armazenada corretamente.

d) Cria automaticamente as pastas necessárias para armazenar as anotações e os arquivos auxiliares.

e) Mantém persistência nos dados salvos, permitindo que o usuário continue a anotação posteriormente sem perder informações.

A estrutura modular implementada garante a separação adequada de responsabilidades entre os componentes da aplicação, assegurando um fluxo consistente de funcionamento.

Dessa forma, a ferramenta desenvolvida atende aos requisitos estabelecidos e proporciona um ambiente intuitivo e eficiente para anotação de imagens. A escolha

de tecnologias modernas como React.js, Tailwind CSS e Vite garantiu um desenvolvimento ágil e um sistema leve, modular e escalável, permitindo futuras expansões. A arquitetura da aplicação foi estruturada para proporcionar uma separação clara entre os componentes, otimizando a interação do usuário e a gestão das anotações.

Ao focar exclusivamente na anotação de bounding boxes no formato YOLO, a ferramenta se alinha às necessidades de modelos de detecção de objetos que priorizam desempenho e simplicidade. A exportação direta dos arquivos no padrão YOLO facilita sua integração com pipelines de treinamento sem a necessidade de conversões adicionais, garantindo praticidade no uso real.

Além disso, a modularidade do projeto permite que futuras iterações incluam novas funcionalidades, como suporte a outros formatos de anotação, melhorias na interface e otimizações no fluxo de trabalho. Com isso, a ferramenta se estabelece como uma base sólida para anotações eficientes, oferecendo um recurso valioso para a construção de datasets de visão computacional.

4 APRESENTAÇÃO DOS RESULTADOS

Neste capítulo, são apresentados os resultados obtidos a partir da utilização da ferramenta desenvolvida, com foco na validação das anotações geradas e sua compatibilidade com plataformas externas. O objetivo é demonstrar a eficácia do sistema, evidenciando seu funcionamento na anotação de imagens, exportação dos dados e validação das caixas delimitadoras em um ambiente de referência, como o Roboflow.

Inicialmente, são descritos os passos realizados para a anotação das imagens na ferramenta, detalhando o processo de criação de caixas delimitadoras e a correta associação das classes aos objetos identificados. Em seguida, é abordada a etapa de exportação das anotações no formato YOLO, garantindo a conformidade dos arquivos gerados com os padrões utilizados para treinamentos em modelos de visão computacional.

Na sequência, são apresentados os procedimentos para upload dos arquivos no Roboflow, destacando a verificação da integridade dos dados e a correspondência entre as anotações realizadas na ferramenta e as exibidas na plataforma externa. A comparação visual das caixas delimitadoras permite confirmar a precisão da exportação e a fidelidade das informações anotadas.

Por fim, os resultados obtidos são analisados, reforçando a contribuição da ferramenta desenvolvida para a anotação eficiente de imagens e sua viabilidade para aplicações em projetos que demandam datasets bem estruturados. Essa análise considera a usabilidade do sistema, a adequação das anotações ao formato esperado e a possibilidade de integração com fluxos de trabalho já estabelecidos no campo da visão computacional.

4.1 Validação Da Ferramenta

A validação da ferramenta desenvolvida tem como objetivo verificar se as anotações realizadas são corretamente geradas no formato YOLO e se essas anotações podem ser utilizadas em uma plataforma externa de referência, Roboflow, para confirmação da integridade e compatibilidade dos dados.

A metodologia de validação consistiu nos seguintes passos:

1. Uso da ferramenta desenvolvida para anotação de imagens.
2. Exportação e verificação dos arquivos YOLO gerados
3. Upload dos arquivos no Roboflow
4. Validação das caixas delimitadoras na plataforma

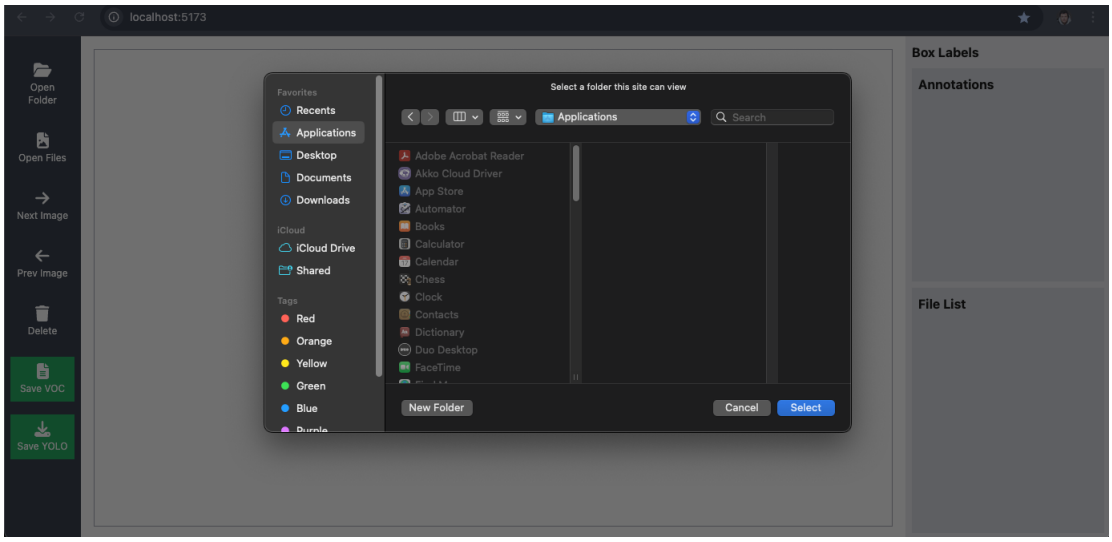
4.1.1 Anotação de Imagens na Ferramenta Desenvolvida

O primeiro passo da validação consistiu na utilização da ferramenta desenvolvida para realizar a anotação manual de um conjunto de imagens. O objetivo foi criar caixas delimitadoras e verificar a correta exportação dessas anotações.

Os passos seguidos foram:

1. Seleção do Diretório com as Imagens O primeiro passo para a validação da ferramenta consiste na seleção da pasta que contém as imagens a serem anotadas, conforme figura 20. O usuário utiliza a interface da ferramenta para abrir um diretório contendo as imagens que serão processadas. Esse processo permite carregar múltiplas imagens ao mesmo tempo, facilitando a navegação e anotação.

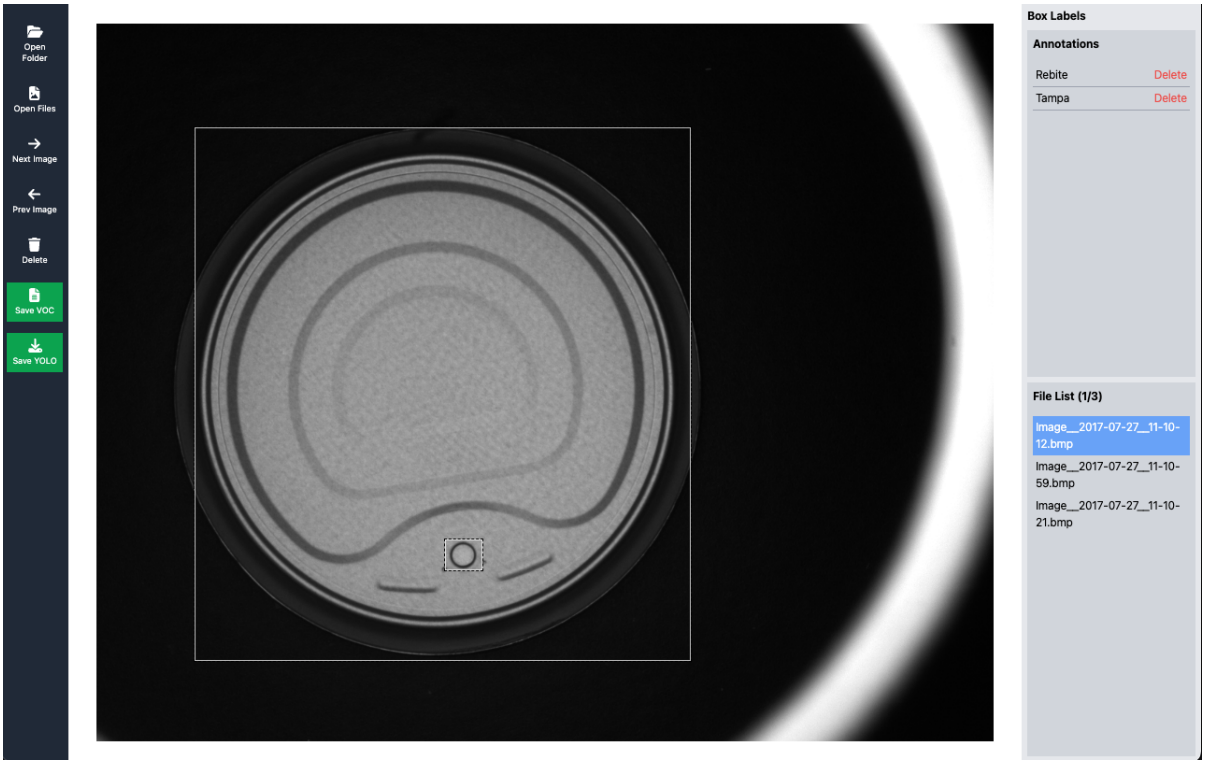
Figura 20 – Seleção da pasta contendo as imagens para anotação



Fonte: Do Autor (2024).

2. Criação das Anotações (Caixas Delimitadoras) Após carregar as imagens, o usuário pode começar a desenhar as caixas delimitadoras ao redor dos objetos de interesse (figura 21). A ferramenta permite adicionar múltiplas anotações por imagem e associar um rótulo a cada bounding box, garantindo que os objetos sejam corretamente identificados.

Figura 21 – Criação das anotações na ferramenta, com caixas delimitadoras e labels

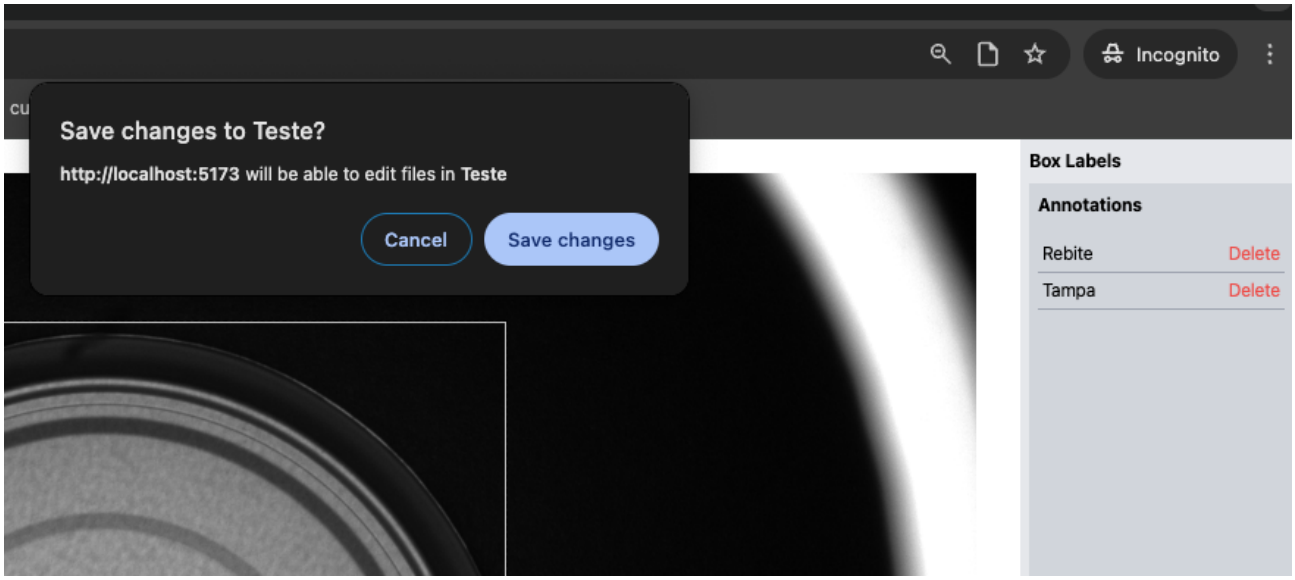


Fonte: Do Autor (2024).

3. Salvando as Anotações

   Após realizar as anotações, o usuário deve salvar os rótulos no formato YOLO. Esse processo cria arquivos .txt correspondentes a cada imagem, contendo as coordenadas das caixas delimitadoras e os respectivos rótulos, como mostra a figura 22.

Figura 22 – Solicitação para salvar as anotações no diretório local

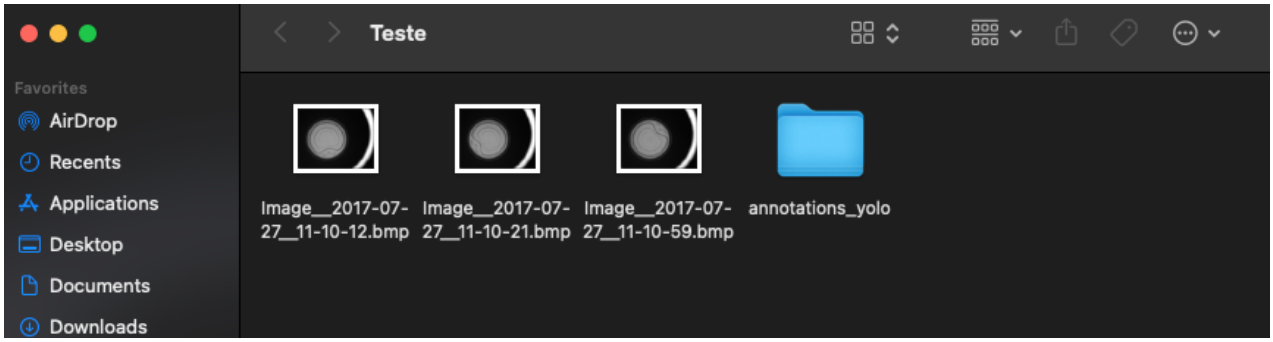


Fonte: Do Autor (2024).

4. Verificação do Diretório de Saída

   Após salvar as anotações, é possível verificar que o diretório contém os arquivos das imagens e a pasta annotations-yolo, onde as anotações são armazenadas no formato YOLO de acordo com a figura 23.

Figura 23 – Verificação da pasta de saída contendo imagens e anotações

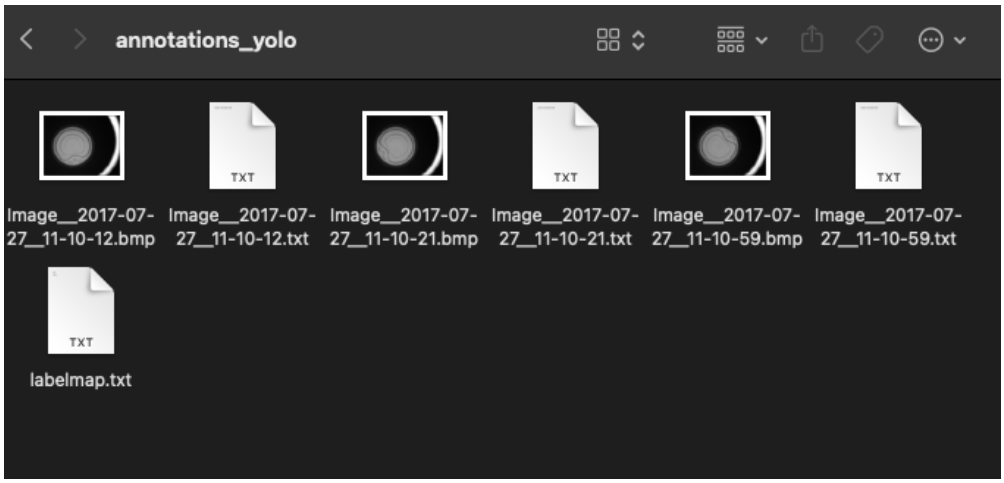


Fonte: Do Autor (2024).

5. Verificação dos Arquivos YOLO

   Dentro da pasta annotations-yolo (figura 24), é possível observar os arquivos .txt gerados para cada imagem. Esses arquivos contêm as coordenadas das caixas delimitadoras e os rótulos das anotações.

Figura 24 – Arquivos YOLO gerados para cada imagem anotada

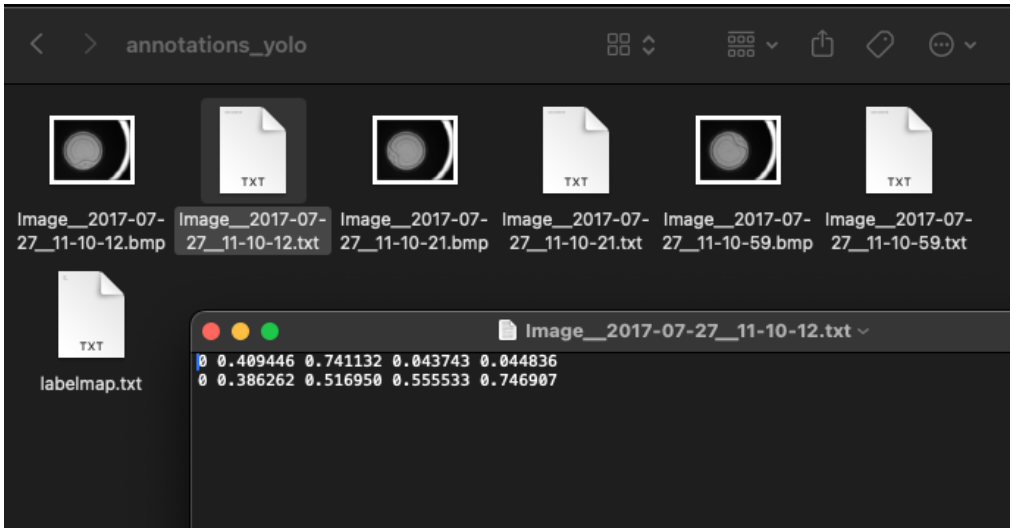


Fonte: Do Autor (2024).

6. Validação do Formato dos Arquivos YOLO

   Por fim, ao abrir um dos arquivos .txt, verifica-se que ele contém os dados estruturados corretamente, com as coordenadas das caixas delimitadoras no formato esperado pelo YOLO, como mostra a figura 25.

Figura 25 – Visualização do conteúdo de um arquivo YOLO gerado



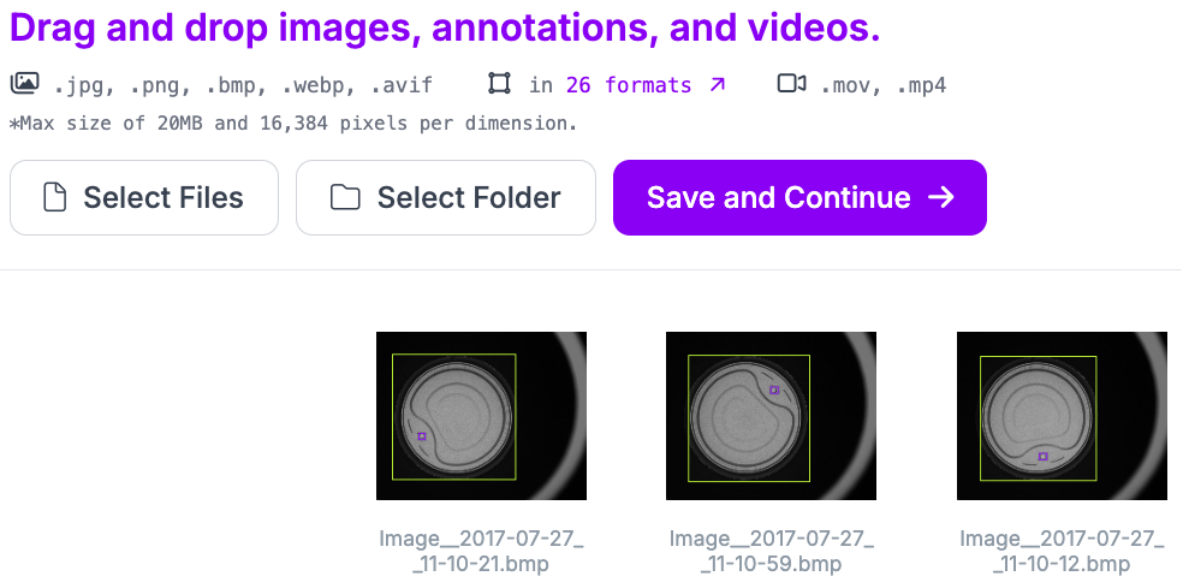
Fonte: Do Autor (2024).

A figura 26 mostra a validação após a anotação, cada imagem gerou um arquivo .txt contendo as coordenadas das caixas delimitadoras normalizadas no formato YOLO.

7. Validação do Formato dos Arquivos YOLO

   Para validar a compatibilidade da ferramenta, os arquivos gerados foram carregados na plataforma Roboflow. A pasta de anotações contendo imagens e arquivos .txt foi selecionada para upload.

Figura 26 – Upload das Anotações no Roboflow

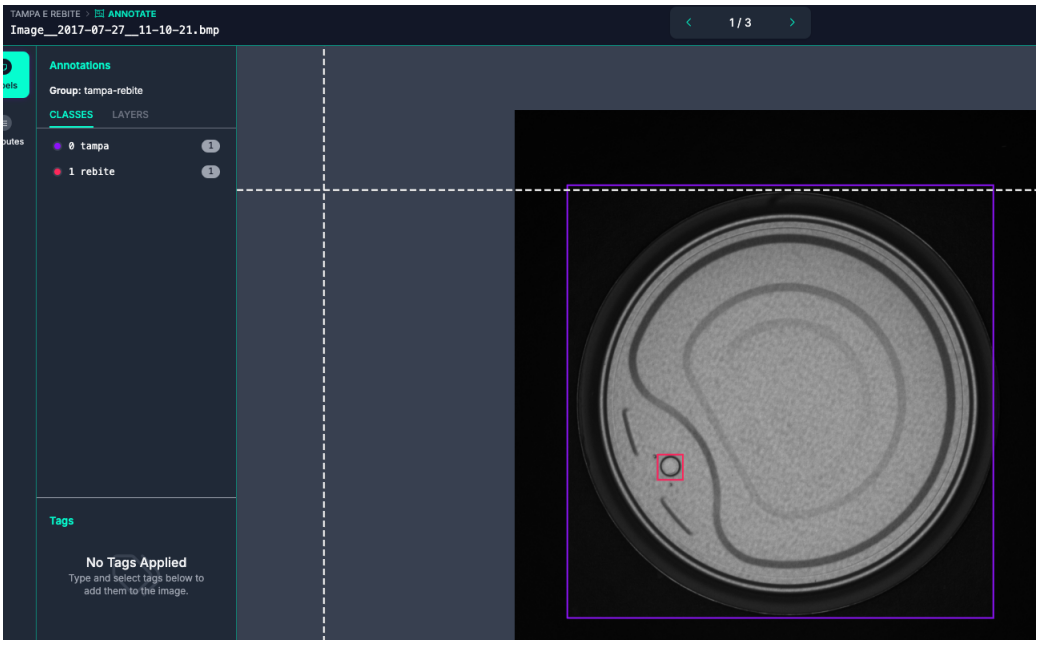


Fonte: Do Autor (2024).

8. Visualização das Anotações no Roboflow

Após o upload, as imagens foram carregadas na interface do Roboflow, e as caixas delimitadoras anotadas foram corretamente identificadas. Isso confirma que a ferramenta desenvolvida gera anotações no formato adequado para integração com plataformas externas, conforme mostra a figura 27.

Figura 27 – Visualização das anotações no Roboflow



Fonte: Do Autor (2024).

4.2 Análise dos Resultados

A validação da ferramenta mostrou que o processo de anotação é realizado corretamente, e os arquivos gerados seguem o padrão esperado pelo formato YOLO. O upload das imagens e anotações no Roboflow confirmou que as caixas delimitadoras foram preservadas e corretamente interpretadas pela plataforma.

1. Compatibilidade com o Formato YOLO: Os arquivos de anotação gerados pela ferramenta seguem corretamente a estrutura exigida pelo formato YOLO. Isso foi confirmado pela leitura das coordenadas das caixas delimitadoras e pela exibição correta das anotações no Roboflow.

2. Facilidade de Uso: O processo de anotação foi intuitivo, permitindo a rápida marcação de objetos nas imagens. A ferramenta possibilita salvar e organizar as anotações de forma eficiente, facilitando sua integração em fluxos de treinamento de modelos de visão computacional.

3. Possibilidades de Expansão: A ferramenta, embora desenvolvida com foco exclusivo na anotação de caixas delimitadoras no formato YOLO, pode ser expandida no futuro para oferecer suporte a outros formatos de anotação, como COCO e

PASCAL VOC. Além disso, a adição de recursos para automação de anotações poderia tornar o processo ainda mais eficiente.

A validação demonstrou que a ferramenta cumpre seu propósito de gerar anotações de caixas delimitadoras no formato YOLO e que os arquivos gerados são compatíveis com plataformas de visão computacional, como o Roboflow. O fluxo de trabalho implementado permite a criação rápida e eficiente de datasets anotados, viabilizando seu uso para treinamento de modelos de detecção de objetos.

Tabela 1 – Cumprimento dos Requisitos

Requisito	**Status**
Permitir anotação manual de imagens	Atendido
Suporte ao formato de exportação YOLO	Atendido
Interface intuitiva para anotação	Atendido
Possibilidade de salvar anotações localmente	Atendido
Compatibilidade com plataformas externas (Roboflow)	Atendido
Facilidade na criação e edição de caixas delimitadoras	Atendido
Suporte a múltiplos formatos de anotação	Não atendido
Integração com banco de dados para armazenamento remoto	Não atendido
Implementação de automação de anotações	Não atendido

Fonte: Autor (2024).

A ferramenta desenvolvida atendeu satisfatoriamente ao objetivo geral, proporcionando uma interface web acessível para anotação de imagens, voltada ao meio acadêmico e à pesquisa em visão computacional. A interface foi projetada para ser intuitiva, facilitando o processo de anotação manual e garantindo compatibilidade com formatos amplamente utilizados, como o YOLO. Além disso, sua estrutura permite futuras integrações, tornando-se um recurso viável para trabalhos posteriores.

No que se refere aos objetivos específicos, os resultados obtidos demonstram um bom nível de cumprimento:

- O estudo dos formatos de anotação foi realizado, possibilitando uma escolha fundamentada na adoção do YOLO como padrão principal, devido à sua ampla aceitação na comunidade de visão computacional.
- Foram selecionadas tecnologias web modernas, como React, para o desenvolvimento da interface, garantindo eficiência, compatibilidade e um ambiente ágil para a anotação de imagens.
- A implementação da ferramenta atendeu às exigências de usabilidade e funcionalidade, permitindo a criação e exportação de anotações de forma prática e precisa.

- A ferramenta foi testada extensivamente, validando seu funcionamento em um ambiente controlado e verificando a geração correta de anotações no formato YOLO.
- O processo de exportação e validação no Roboflow comprovou que os arquivos gerados são compatíveis e interpretados corretamente por plataformas externas, demonstrando a integridade dos dados anotados.
- A disponibilização em um repositório público ainda não foi concluída, mas a documentação do sistema já foi elaborada, registrando decisões técnicas e melhorias futuras.
- A documentação do desenvolvimento foi mantida ao longo do projeto, garantindo que todas as decisões técnicas, desafios enfrentados e soluções implementadas fossem registradas.

A ferramenta desenvolvida foi disponibilizada publicamente no GitHub, garantindo acesso aberto à comunidade acadêmica e de pesquisa em visão computacional. O repositório inclui tanto o código-fonte quanto a documentação, permitindo que outros desenvolvedores utilizem, contribuam e expandam suas funcionalidades. Essa abordagem reforça um dos principais objetivos do projeto, que é tornar a anotação de imagens mais acessível e colaborativa. O código pode ser acessado em: https://github.com/geromuller/image-annotation-tool

Com base nos resultados obtidos, a ferramenta pode ser considerada uma solução funcional para a anotação de imagens em aplicações de visão computacional. Futuras melhorias podem incluir a implementação de novos métodos de anotação e a otimização da interface para tornar o processo ainda mais ágil.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento desta ferramenta de anotação de imagens teve como objetivo fornecer uma solução eficiente e prática para a marcação de bounding boxes no formato YOLO. A partir da revisão teórica, do estudo das ferramentas existentes e da implementação da solução proposta, foi possível criar um sistema funcional que atende às necessidades de anotação de imagens de forma rápida, intuitiva e acessível. A validação realizada confirmou a eficácia da ferramenta, demonstrando que as anotações geradas são corretamente interpretadas por plataformas externas, como o Roboflow.

Ao longo deste trabalho, foram abordadas diversas etapas do processo de anotação de imagens, desde a importância da rotulagem para o treinamento de modelos de visão computacional até a análise de ferramentas já consolidadas no mercado. A comparação com softwares como LabelImg demonstrou que a ferramenta desenvolvida consegue suprir uma necessidade específica do fluxo de trabalho de pesquisadores e desenvolvedores que utilizam o formato YOLO. Diferente das demais soluções, que oferecem múltiplas funcionalidades e, muitas vezes, exigem um ambiente mais robusto para execução, a ferramenta proposta neste estudo prioriza simplicidade e eficiência, eliminando elementos que poderiam tornar o processo mais complexo para usuários que precisam de uma anotação direta e rápida.

A validação foi conduzida por meio de um fluxo estruturado de teste, no qual imagens foram anotadas utilizando a ferramenta e posteriormente carregadas no Roboflow para verificação da integridade das bounding boxes. Os resultados indicaram que todas as anotações foram corretamente interpretadas pela plataforma, sem desvios de posicionamento ou erros na atribuição de classes. Esse resultado confirma que a ferramenta atendeu ao requisito principal de compatibilidade e precisão, demonstrando ser uma alternativa viável para pesquisadores e profissionais que trabalham com datasets no formato YOLO.

5.1 Contribuições do Trabalho

Este trabalho trouxe contribuições relevantes para a área de anotação de imagens, especialmente no contexto de detecção de objetos utilizando modelos de deep learning. As principais contribuições podem ser destacadas da seguinte forma:

1. Desenvolvimento de uma ferramenta especializada para anotação YOLO: Diferente de outras soluções que suportam múltiplos formatos e possuem interfaces mais complexas, a ferramenta desenvolvida foca exclusivamente no formato YOLO, garantindo um fluxo de trabalho otimizado para esse tipo de anotação.

2. Facilidade de uso e acessibilidade: A ferramenta foi projetada para ser executada diretamente em navegadores modernos, sem necessidade de instalação de pacotes adicionais, tornando o processo de anotação mais acessível a usuários de diferentes níveis de experiência.

3. Validação prática e compatibilidade com ferramentas externas: A integração bem-sucedida com o Roboflow demonstrou que a ferramenta gera anotações corretas e compatíveis com pipelines de treinamento, reforçando sua aplicabilidade em projetos reais.

4. Código modular e expansível: A estrutura da aplicação foi organizada de maneira modular, permitindo que futuras melhorias possam ser incorporadas sem comprometer a funcionalidade principal.

Além dessas contribuições, o trabalho também reforça a importância da anotação manual no treinamento de modelos de visão computacional. Mesmo com avanços na área de aprendizado semi-supervisionado e autoanotação, a intervenção humana ainda desempenha um papel fundamental na construção de datasets de alta qualidade. Dessa forma, ferramentas eficientes e especializadas, como a desenvolvida neste projeto, continuam sendo essenciais para o avanço da inteligência artificial aplicada à visão computacional.

5.2 Limitações e Desafios

Embora a ferramenta tenha cumprido seu propósito, algumas limitações foram identificadas e podem servir como base para trabalhos futuros:

1. Suporte apenas a bounding boxes: Atualmente, a ferramenta não oferece suporte a métodos mais avançados de anotação, como segmentação semântica ou keypoints. Isso a torna adequada apenas para tarefas de detecção de objetos, limitando seu uso em aplicações que exigem maior granularidade na anotação.

2. Falta de automação no processo de anotação: Algumas ferramentas modernas, como CVAT e Roboflow, possuem funcionalidades de pré-anotação utilizando modelos pré-treinados, o que acelera o processo de rotulagem. A ferramenta desenvolvida não inclui esse recurso, sendo totalmente manual.

3. Execução exclusivamente local: Atualmente, a ferramenta é executada diretamente no navegador, sem suporte para armazenamento em nuvem ou compartilhamento de anotações entre múltiplos usuários. Essa característica pode limitar seu uso em projetos colaborativos.

Apesar dessas limitações, o desenvolvimento da ferramenta cumpriu seu objetivo central, validando a abordagem adotada e garantindo que as anotações fossem corretamente interpretadas por frameworks externos.

5.3 Trabalhos Futuros

Dada a base sólida construída neste trabalho, algumas melhorias podem ser implementadas em versões futuras da ferramenta. Algumas direções possíveis para aprimoramento incluem:

1. Expansão para novos formatos de anotação: Embora o foco tenha sido no formato YOLO, a inclusão de suporte a outros formatos, como COCO e PASCAL VOC, pode tornar a ferramenta mais versátil.
2. Implementação de recursos de automação: A incorporação de modelos de pré-anotação pode reduzir o esforço manual e agilizar o processo de rotulagem, tornando a ferramenta mais eficiente.
3. Integração com plataformas de armazenamento em nuvem: Implementar suporte para sincronização com serviços como Google Drive ou AWS S3 pode permitir que usuários armazenem e compartilhem anotações facilmente.
4. Aprimoramento da interface do usuário: Melhorias na interface gráfica podem tornar a experiência do usuário ainda mais intuitiva, especialmente para aqueles sem experiência prévia em anotação de imagens.

Essas melhorias podem tornar a ferramenta ainda mais útil e consolidá-la como uma alternativa robusta para anotação de imagens no contexto de visão computacional.

REFERÊNCIAS

CAO, Z. *et al.* Realtime multi-person 2d pose estimation using part affinity fields. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 7291–7299, 2017. 32

CORPORATION, I. Cvat: Computer vision annotation tool. *GitHub Repository*, 2019. Disponível em: https://github.com/openvinotoolkit/cvat. 38

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2005. p. 886–893. 20

FORSYTH, D. A.; PONCE, J. ***Computer Vision: A Modern Approach***. [S.l.]: Prentice Hall Professional Technical Reference, 2002. 15, 17

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4th. ed. [S.l.]: Pearson, 2018. ISBN 978-0133356724. 18

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. ***Deep learning***. [S.l.]: MIT press, 2016. 24

KRIZHEVSKY SUTSKEVER, H. ***ImageNet Classification with Deep Convolutional Neural Networks***. [S.l.: s.n.], 2012. 16

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep Learning**. *Nature*, v. 521, p. 436–44, 05 2015. 12

LIU, W. *et al.* Ssd: Single shot multibox detector. In: *European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2016. p. 21–37. 21, 30

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 3431–3440, 2015. 31

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, v. 60, n. 2, p. 91–110, 2004. 19

REDMON, J. *et al.* You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 779–788. 20, 30

REN, S. *et al.* Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2015. p. 91–99. 20

ROBOFLOW. ***Roboflow: Turn Images Into Deployable Computer Vision Models***. 2021. https://roboflow.com/. 35, 38

RUSSAKOVSKY, O. *et al.* **ImageNet Large Scale Visual Recognition Challenge**. *International Journal of Computer Vision*, v. 115, p. 211 – 252, 2014. 12

SAGER, C. J. C.; ZSCHECH, P. **A survey of image labelling for computer vision applications**. Taylor Francis, v. 4, 2021. 13

SZELISKI, R. ***Computer Vision: Algorithms and Applications***. [S.l.]: Springer Science Business Media, 2010. 12, 15

TZUTALIN. ***LabelImg***. 2015. https://github.com/tzutalin/labelImg. GitHub repository. 34, 38

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2001. p. 511–518. 19