**UNIVERSIDADE FEDERAL DO CEARÁ**

**CENTRO DE TECNOLOGIA**

**DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA**

**CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO**

**TÁCIO SOARES AGUIAR**

# VISÃO COMPUTACIONAL PARA DETECÇÃO E SEGMENTAÇÃO DE RESÍDUOS SÓLIDOS URBANOS

**FORTALEZA**

**2024**

TÁCIO SOARES AGUIAR

# VISÃO COMPUTACIONAL PARA DETECÇÃO E SEGMENTAÇÃO DE RESÍDUOS SÓLIDOS URBANOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. José Gilvan Rodrigues Maia

FORTALEZA

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

A233v Aguiar, Tácio Soares.
Visão computacional para detecção e segmentação de resíduos sólidos urbanos / Tácio Soares Aguiar. – 2024.
55 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia de Computação, Fortaleza, 2024.
Orientação: Prof. Dr. José Gilvan Rodrigues Maia.

1. Visão computacional . 2. Detecção de objetos. 3. Gerenciamento de lixo urbano. 4. Resíduos sólidos urbanos. 5. Segmentação de objetos. I. Título.

CDD 621.39

TÁCIO SOARES AGUIAR

# VISÃO COMPUTACIONAL PARA DETECÇÃO E SEGMENTAÇÃO DE RESÍDUOS SÓLIDOS URBANOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: 26 de setembro de 2024

BANCA EXAMINADORA

---

Prof. Dr. José Gilvan Rodrigues Maia (Orientador)
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Paulo Antônio Leal Rego
Universidade Federal do Ceará (UFC)

---

Prof. Me. Artur de Oliveira da Rocha Franco
Universidade Federal do Ceará (UFC)

## AGRADECIMENTOS

Gostaria de agradecer primeiramente aos meus pais e meu irmão, por todo o amor, apoio e força que sempre me deram e por acreditarem em mim em todos os momentos. À Andressa, pela paciência, carinho e incentivos constantes, que me motivaram a seguir em frente estando presente em cada conquista. À minha tia Cisinha, por ter me apoiado em um momento importante da minha vida, com sua gentileza, compreensão e afeto.

Ao professor Gilvan Maia, meu orientador, por sua dedicação, atenção e pelos ensinamentos valiosos que foram fundamentais não apenas para o desenvolvimento deste trabalho, mas também para minha formação pessoal e profissional.

Ao professor Paulo Rego, por sua ajuda em diversas fases deste trabalho, cuja colaboração foi essencial para que eu pudesse atingir os resultados aqui apresentados.

Aos professores do Departamento de Engenharia e Tecnologia da Informação (DETI), por proporcionarem um ensino de excelência e por contribuírem significativamente para a minha formação acadêmica. O conhecimento e a experiência de cada um de vocês foram determinantes para o meu crescimento.

A todos, meu sincero muito obrigado.

“A comparação é a ladra da alegria.”

(Theodore Roosevelt)

## RESUMO

O gerenciamento adequado do lixo urbano é um desafio global devido ao crescente volume de resíduos gerados diariamente e ao impacto negativo que isso causa na saúde humana e no meio ambiente. De acordo com a Organização das Nações Unidas (ONU), a grande maioria dos produtos que consumimos são descartados num curto período de tempo, resultando em uma quantidade alarmante e desordenada de resíduos. Além disso, a demanda crescente por recursos e o descarte inadequado de lixo contribuem para a degradação do meio ambiente. Este estudo propõe o uso de técnicas de Visão Computacional com o objetivo de detectar e segmentar diferentes tipos de lixo com foco em ambientes urbanos. O modelo de *Deep Learning* proposto foi capaz de produzir resultados compatíveis com o estado da arte, inclusive superando os demais métodos analisados para uma coleção de imagens que representa cenários urbanos, mAP 65,9%. Muito embora o modelo proposto tenha se mostrado limitado na detecção de objetos pequenos e em outros tipos de cenários.



**Palavras-chave:** Visão Computacional. Detecção de objetos. Segmentação de Objetos. Gerenciamento de Lixo Urbano. Resíduos Sólidos Urbanos

## ABSTRACT

Proper management of urban waste is a global challenge due to the growing volume of waste generated daily and the negative impact this has on human health and the environment. According to the UN, most products we consume are discarded quickly, resulting in an alarming and disorganized amount of waste. In addition, the growing demand for resources and improper waste disposal contribute to environmental degradation. This study proposes using Computer Vision techniques to detect and segment different types of waste in urban environments. The proposed *Deep Learning* model produced results compatible with state-of-the-art and surpassed the performance of other methods analyzed regarding a collection of images representing urban scenarios, mAP 65,9%. On the other hand, the proposed model could not properly handle small objects and non-urban scenarios.



**Keywords:** Computer Vision. Object detection. Object segmentation. Urban waste management. Urban Solid Waste

## LISTA DE FIGURAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AP | *Average Precision* |
| FN | *False Negative* |
| FP | *False Positive* |
| FPN | *Feature Pyramid Network* |
| IoU | *Intersection over Union* |
| mAP | *Mean Average Precision* |
| ONU | Organização das Nações Unidas |
| PAN | *Path Aggregation Network* |
| ROV | *Remotely Operated Vehicle* |
| RSU | Resíduos Sólidos Urbanos |
| SAHI | *Slicing Aided Hyper Inference* |
| SAM | *Segment Anything Model* |
| TACO | *Trash Annotations in Context for Litter Detection* |
| TP | *True Positive* |
| YOLO | *You Only Look Once* |

**SUMÁRIO**

# 1 INTRODUÇÃO

O presente trabalho versa sobre Resíduos Sólidos Urbanos (RSU), que normalmente são chamados de lixo. A produção desenfreada de resíduos sólidos nas cidades torna-se um grande transtorno para a sociedade urbana e o grande consumo de produtos diversos com o descarte inadequado acarreta fatores diversos indesejáveis para a sociedade e o meio ambiente (CARDOSO; CARDOSO, 2016).

Entre os impactos ambientais negativos que podem ser originados a partir do lixo urbano produzido estão os efeitos decorrentes da prática de descarte e disposição inadequada de resíduos sólidos, como às margens de ruas ou flúmens. Essas práticas habituais podem provocar, entre outras coisas, contaminação de corpos d'água, assoreamento, enchentes, proliferação de vetores transmissores de doenças, tais como cães, gatos, ratos, baratas, moscas, vermes, entre outros. Some-se a isso a poluição visual, mau cheiro e contaminação do ambiente (MUCELIN CARLOS ALBERTO BELLINI, 2008).

## 1.1 Contextualização

Além do impacto ambiental o lixo urbano traz custos constantes para todos, principalmente para grandes cidades que necessitam do serviço privado para atender essa demanda, estima-se que capitais do Nordeste possuam um custo *per capta* de R$ 151,23 em média por ano de gastos com toda cadeia de cuidados com RSU (RODRIGUES *et al.*, 2016).

Esses problemas levaram a Prefeitura da cidade de Fortaleza–CE a tomar medidas públicas, criando uma taxa do lixo para garantir a universalização do saneamento básico para sua população (Samuel Pinusa, 2022). Além disso, os descartes irregulares de lixo realizados pelos moradores prejudicam tanto o meio ambiente como a saúde pública. A Prefeitura de Fortaleza, ainda na tentativa de coibir tal prática, estabelece uma multa mínima diária no valor de R$ 389,39 àqueles que forem flagrados realizando algum tipo de descarte. Porém, a falta de fiscalização dificulta tal ação (GOMES; BELéM, 2022).

Essa falta de fiscalização pode ser contornada com o objeto de estudo desse trabalho, utilizando todo o sistema de monitoramento urbano já instalado nas cidades é possível captar imagens das ruas praticamente vinte e quatro horas por dia. Ao juntar essas imagens capturadas, podemos utilizar várias técnicas de visão computacional como Redes Neurais Convolucionais (ANKILE *et al.*, 2020), Segmentação Semântica (KIRILLOV *et al.*, 2023) entre outras, para

gerar *insights* que ajudem na tomada de decisão por parte da Prefeitura, por exemplo, nesse caso da multa mínima diária.

Com a implementação de um sistema inteligente de gerenciamento de lixo urbano, surgem diversas possibilidades que podem otimizar a coleta e o monitoramento dos resíduos. Uma dessas possibilidades é a coleta de lixo mais eficiente, utilizando os dados das classificações dos tipos de lixo e sua localização. Com base nessas informações, é possível identificar o momento em que determinado material foi depositado, permitindo um acionamento rápido da coleta e evitando o acúmulo de resíduos.

Além disso, o sistema inteligente de monitoramento pode proporcionar uma série de benefícios na gestão do lixo urbano. Ele pode ajudar a identificar áreas com maior incidência de descarte irregular, possibilitando ações mais direcionadas de fiscalização e conscientização. Também pode contribuir para a implementação de políticas públicas mais eficazes e justas, utilizando as informações obtidas para a criação de estratégias de taxação mais adequadas e direcionadas aos locais e tipos de lixo específicos.

Em conclusão, o presente trabalho busca desenvolver modelos que possam fazer uso dessas tecnologias de visão computacional para verificar a possibilidade de melhorar o gerenciamento do lixo urbano.

## 1.2 Objetivos

### *1.2.1 Objetivo Geral*

Esse trabalho tem como objetivo geral desenvolver e avaliar uma técnica de *machine learning* para detectar e segmentar diferentes tipos de lixos em áreas urbanas.

### *1.2.2 Objetivos Específicos*

Os objetivos específicos deste trabalho são os seguintes:

- Realizar uma revisão do estado da arte em pesquisas relacionadas, a fim de compreender as abordagens e técnicas existentes na área de identificação e classificação de lixo em ambientes urbanos;
- Propor ou adaptar uma técnica adequada para o desenvolvimento do modelo de *Machine learning*, levando em consideração as características específicas do problema;
- Realizar a coleta de dados, obtendo imagens de diferentes tipos de lixo em áreas urbanas,

a fim de montar um *dataset* apropriado para as etapas da experimentação;

- Conduzir experimentos comparativos utilizando diferentes modelos, adotando para tanto o conjunto de dados personalizado; e
- Avaliar o desempenho dos modelos utilizando métricas de avaliação de modelos de visão computacional.

## 1.3 Organização da Monografia

O restante deste trabalho está organizado da seguinte maneira:

- O Capítulo 2 trata do referencial teórico, oferecendo uma visão abrangente dos principais conceitos e teorias em Visão Computacional, com foco nos temas mais relevantes para o desenvolvimento deste trabalho;
- O Capítulo 3 explora os principais conjuntos de dados encontrados na literatura sobre Visão Computacional aplicados à detecção e segmentação de lixo;
- No Capítulo 4, são discutidos os trabalhos relacionados que apresentam maior correlação com a proposta deste estudo no sentido de detectar e segmentar lixo em imagens digitais;
- O Capítulo 5 detalha a metodologia empregada e as decisões tomadas ao longo do desenvolvimento da pesquisa;
- O Capítulo 6 apresenta e discute os resultados obtidos durante a execução do trabalho; e
- Por fim, o Capítulo 7 apresenta as considerações finais sobre o presente trabalho e aponta oportunidades para trabalhos futuros.

# 2 REFERENCIAL TEÓRICO

É um tema muito abrangente e tecnicamente complexo, portanto o foco deste capítulo é prover uma visão geral dos assuntos e teorias envolvidos na visão computacional que estejam preferencialmente relacionados ao desenvolvimento da proposta deste trabalho. Isso significa que alguns pormenores não serão abordados e que o leitor interessado pode recorrer a outras leituras para uma abordagem mais ampla e profunda:

- Já sobre Processamento Digital de Imagens, recomenda-se a leitura das edições mais recentes do livro de Gonzalez (2009) e Jain (1989);
- Sobre Aprendizagem Profunda, o artigo de LeCun *et al.* (2015) é um bom ponto de partida para entender os princípios de funcionamento, suas aplicações e como esse paradigma de aprendizagem revolucionou diversos campos. Os livros de Goodfellow *et al.* (2016) e Zhang *et al.* (2023) são os materiais mais recomendados para uma formação mais completa nesses assuntos; e
- Por fim, sobre Visão Computacional, recomendam-se os livros de Stockman e Shapiro (2001), Hartley e Zisserman (2003) e Szeliski (2022).

Visão Computacional pode ser definida como o campo de estudo focado em, de uma forma geral, fazer com que os computadores "vejam". É um campo multidisciplinar considerado como uma subárea da Inteligência Artificial e Aprendizado de Máquina, e seu principal objetivo é entender o conteúdo de imagens digitais. Essa tarefa normalmente possui como abordagem a tentativa de reproduzir a capacidade de visão humana, onde o modelo terá que extrair a descrição da imagem para entender seu conteúdo (BROWNLEE, 2019).

## 2.1 Detecção de Objetos em Imagens Digitais

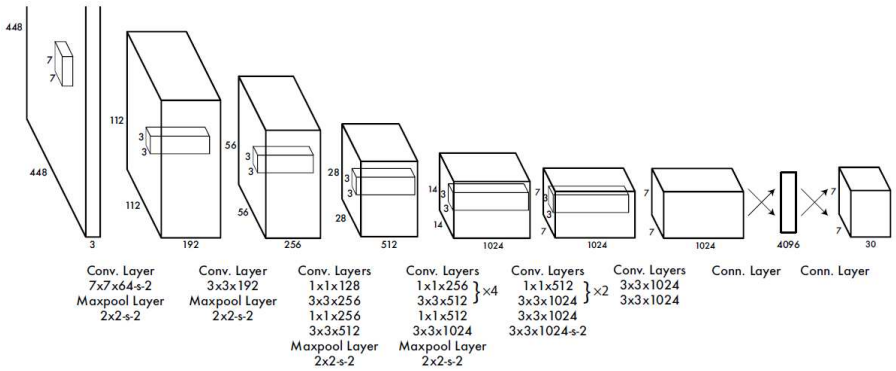
A detecção dos objetos que aparecem em uma imagem digital é uma tarefa bem estudada no campo da Visão Computacional. Sua principal tarefa é determinar se existe alguma instância de um objeto de interesse em uma imagem, geralmente a partir de uma ou mais categorias de objetos (e.g., pessoa, carro, moto, casa, gato, cachorro). Caso tal objeto esteja presente, retorna-se a configuração espacial (e.g., coordenadas, orientação, dimensões, etc) e instância de cada objeto detectado (RUSSAKOVSKY *et al.*, 2015). É possível resolver diversos problemas complexos no campo da Detecção de Objetos, como segmentação, *tracking* de objetos

e detecção de eventos (GONZALEZ, 2009). Esses avanços oferecem suporte a uma ampla gama de aplicações, abrangendo desde visão robótica, eletrônicos de consumo e segurança até áreas mais específicas, como direção autônoma, interação homem-computador, recuperação de imagens baseada em conteúdo, vigilância por vídeo inteligente e realidade aumentada (SZELISKI, 2022). Essa flexibilidade e aplicabilidade tornam essas técnicas essenciais para o desenvolvimento de soluções inovadoras em diferentes setores.

### *2.1.1 YOLO*

YOLO (REDMON *et al.*, 2016) é um algoritmo de detecção de objetos projetado para operar em tempo real e que pode identificar objetos em uma imagem. As informações retornadas por esse algoritmo permitem desenhar uma caixa em volta do objeto identificado, que chamaremos de *bounding box*, e designar uma classe de acordo com sua probabilidade estimada.

Figura 1 – Arquitetura do YOLO contendo 24 camadas convolucionais seguidas de 2 camadas totalmente conectadas




Fonte: Redmon *et al.* (2016)

Nesse estudo, os autores utilizaram as primeiras 20 camadas convolucionais do *backbone* da rede e acrescentaram uma camada de *pooling* média e uma camada totalmente conectada (vide Figura 1). Essa arquitetura foi então pré-treinada e validada usando o conjunto de dados ImageNet 2012 (KRIZHEVSKY *et al.*, 2012). Durante a inferência, as quatro últimas camadas e as duas camadas totalmente conectadas são integradas à rede para a realização das predições. O otimizador *Stochastic Gradient Descent* (KIEFER; WOLFOWITZ, 1952) é tipicamente utilizado durante o processo de treinamento do YOLO, o qual calcula os pesos dos neurônios da rede neural . Nesse trabalho, autores também fizeram uma atualização da função de perda utilizada no modelo:

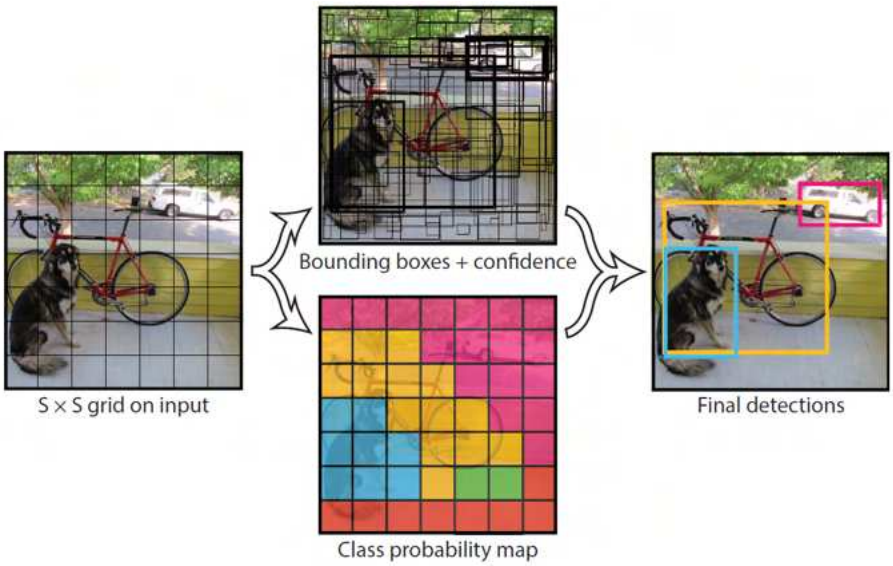
$$
\begin{aligned}
&\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
+\lambda_{coord} &\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left( C_i - \hat{C}_i \right)^2 \\
&+\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left( C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
\tag{2.1}
$$

- $\sum_{i=0}^{S^2} \sum_{j=0}^{B}$ - Soma a perda total de todas as $S^2$ células da grade e para cada uma das $B$ *bounding boxes* por célula, permitindo que o YOLO detecte vários objetos em diferentes posições, podendo prever múltiplas *boxes* por célula;
- $\lambda_{coord}$ - Hiper-parâmetro responsável por ajustar a importância relativa da perda das coordenadas da *bounding box* em comparação com outras partes da função de perda, um valor maior desse parâmetro acarretará aumento do peso dado àquela localização do objeto;
- $1_i j^{\text{obj}}$ - É um indicador binário que vale 1 se a *bounding box* $j$ da célula $i$ tiver um objeto e 0 caso não tenha. Garantindo que a parte da perda relacionada às coordenadas só será aplicada para as células que de fato contêm um objeto;
- $\left(x_i - \hat{x}_i\right)^2, (y_i - \hat{y}_i)^2$ - Mede o erro entre as coordenadas reais ($x_i$, $y_i$) do centro da *bounding box* do objeto e as coordenadas previstas ($\hat{x}_i$, $\hat{y}_i$) pelo YOLO. Garantindo que o modelo aprenda a prever a posição do objeto dentro da célula correspondente;
- $(\sqrt{w_i} - \sqrt{\hat{w}_i})^2, (\sqrt{h_i} - \sqrt{\hat{h}_i})^2$ - Mede o erro entre as dimensões reais da *bounding box* (largura $w_i$ e altura $h_i$). A raiz quadrada ajuda na estabilização da variação, prevenindo que grandes mudanças dominem a função de perda, ajudando na robustez das previsões das *bounding boxes*;
- $\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left( C_i - \hat{C}_i \right)^2$ - Mede o erro de confiança para as *boxes* que contêm objetos. O termo $C_i$ é a confiança prevista pelo modelo;
- $\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} \left( C_i - \hat{C}_i \right)^2$ - Calcula o erro de confiança para as células que não contêm objeto. O termo $\lambda_{noobj}$ ajusta o peso de acordo com essa penalidade, reduzindo a quantidade de falsos positivos;
- $\sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$ - Mede o erro da classificação, ou seja, a diferença entre as

probabilidades previstas $\hat{p}_i(c)$ e as reais $p_i(c)$, visando minimizar o erro de classificação para ajudar o modelo a prever corretamente a classe do objeto presente na célula *i*.

Essa função de perda vai ajudar o treinamento do modelo YOLO a produzir um resultado capaz localizar o objeto pelas *bounding boxes* e classificá-los corretamente em categorias, ajustando seus parâmetros de forma os erros dessas duas tarefas sejam minimizados durante o processo de otimização subjacente ao treinamento.

Figura 2 – Demonstração do processo de classificação de objetos do YOLO

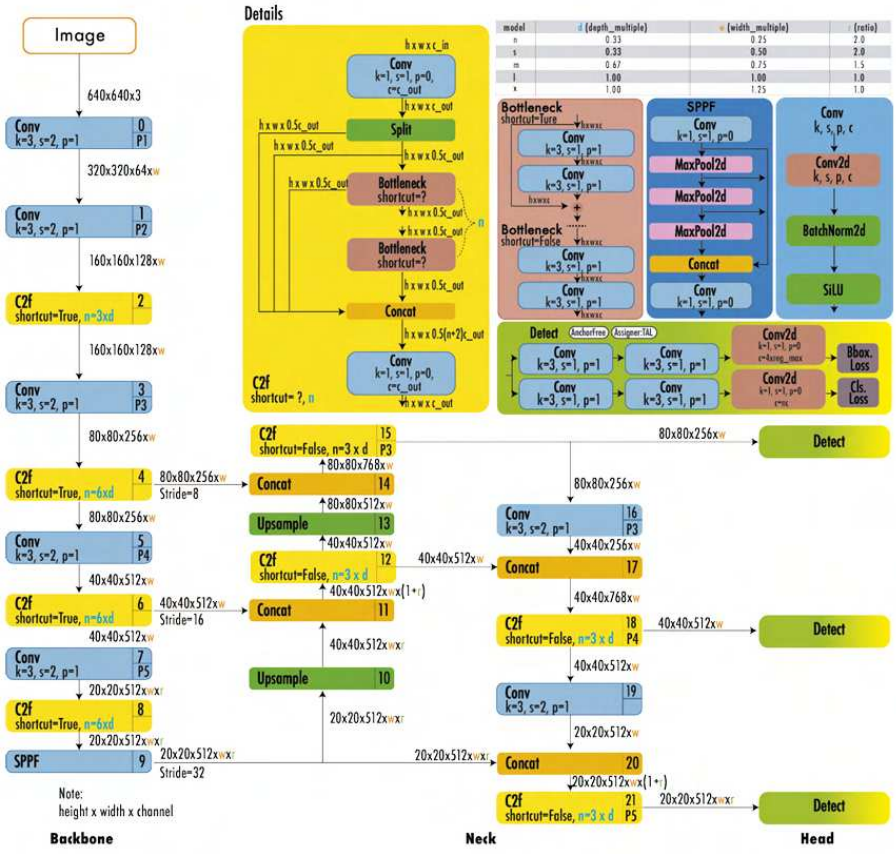



Fonte: Redmon *et al.* (2016).

### *2.1.2 YoloV8*

O YOLOV8 (JOCHER *et al.*, 2023), uma versão mais recente em relação ao estudo original do (REDMON *et al.*, 2016), traz uma nova arquitetura de rede neural que utiliza *Feature Pyramid Network* (FPN) e *Path Aggregation Network* (PAN). O FPN reduz gradualmente a resolução espacial da imagem de entrada, ao mesmo tempo que aumenta o número de canais de características, criando assim um mapa de características que permite detectar objetos em diferentes escalas e resoluções. Já a arquitetura FPN combina informações de diferentes níveis da rede, conectando camadas para melhorar a captura de características em várias escalas e resoluções, o que contribui para uma maior precisão na detecção de objetos de diferentes tamanhos e formatos (TERVEN *et al.*, 2023). YOLOv8 usa CIoU (ZHENG *et al.*, 2020) e DFL (LI *et al.*, 2020) como funções de perda para as *bounding boxes* e *binary cross-entropy* para a função de perda de classificação.

Figura 3 – Arquitetura do YOLOV8




Fonte: Terven *et al.* (2023).

## 2.2 Segmentação Semântica

A Segmentação Semântica é uma difícil tarefa na área da Visão Computacional, mas nos últimos anos, o desempenho desse desafio tem melhorado através da utilização de técnicas de *deep learning* (HAO *et al.*, 2020). Ao analisar uma imagem, atribui uma categoria a cada *pixel* presente. Por conta disso a Segmentação Semântica é capaz de fornecer informações de categoria ao nível do *pixel*, então, muitas aplicações do mundo real beneficiam desta tarefa, como os veículos autônomos (HA *et al.*, 2017), a detecção de pedestres (LIU; STATHAKI, 2018), a detecção de defeitos (XU *et al.*, 2022). Hao *et al.* (2020) afirma que a informação semântica ao nível do *pixel* permite que os modelos aprendam sobre posições espaciais ou a fazer julgamentos importantes. Por conta desse detalhe, a segmentação semântica distingue-se de outras tarefas comuns de visão computacional.

No artigo de Kirillov *et al.* (2023) foi introduzido o SAM, e no ano seguinte Ravi *et al.* (2024) foi publicada sua segunda versão, que é considerada como o estado da arte no momento da escrita desta monografia. Na Figura 4 é possível visualizar o resultado da segmentação realizada em vídeos via SAM.

Figura 4 – SAM2 segmentando *frames* de vídeos.

Fonte: Ravi *et al.* (2024).

## 2.3 Medindo o Desempenho de Modelos YOLO

Os modelos da família YOLO oferecem uma gama de métricas úteis tanto para a etapa de *fine-tuning* quanto na comparação entre modelos, seja eles da própria família YOLO ou de outras arquiteturas. Essa sessão fará um apanhado geral desses conceitos que serão utilizados ao longo do trabalho.

### *2.3.1 Precisão e Recall*

Precisão e *Recall* são as métricas mais utilizadas para avaliar o desempenho de modelos com tarefa de reconhecimento de padrões (PAGANI *et al.*, 2018). Essas métricas são pontuações de 0 a 1, comumente vistas em porcentagem, onde P se refere a um conjunto dos itens previstos e G se refere ao conjunto *ground truth*, que é o conjunto com informações confiáveis. O comportamento desses dois conjuntos é representado na Figura 5, *False Negative* (FN) é a parte do conjunto que foi identificada como uma amostra negativa quando, na verdade deveria ser positiva. *False Positive* (FP), por outro lado foi identificado como positivo, mas, na verdade, é uma amostra negativa. *True Positive* (TP) é o conjunto identificado como positivo e é positivo (FRÄNTI; MARIESCU-ISTODOR, 2023).

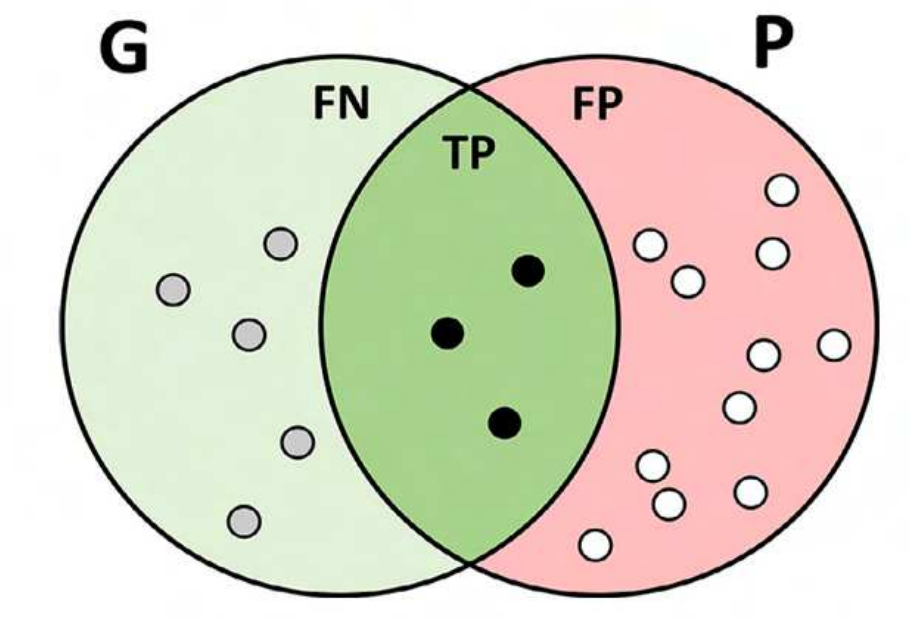
Fränti e Mariescu-Istodor (2023) demonstra esse conceito pelas seguintes fórmulas:

$$Precisao = \frac{|G \bigcap P|}{|P|} \tag{2.2}$$

$$Recall = \frac{|G \bigcap P|}{|G|} \tag{2.3}$$

A atuação do modelo analisada pela Precisão e *Recall* deve ser observada como um *trade-off*, onde o objetivo da aplicação deve ser levado em consideração no momento do *fine-tuning*.

Figura 5 – Comportamento




Fonte: Fränti e Mariescu-Istodor (2023).

#### *2.3.1.1 F1 Score e AP*

$$F1 = 2 \times \frac{precisao \times recall}{precisao + recall} \tag{2.4}$$

Hicks *et al.* (2022) diz que a métrica F1 ou apenas F é a média harmônica para duas classes, nesse caso, Precisão e *Recall*, que penaliza os valores extremos das duas medidas igualmente.

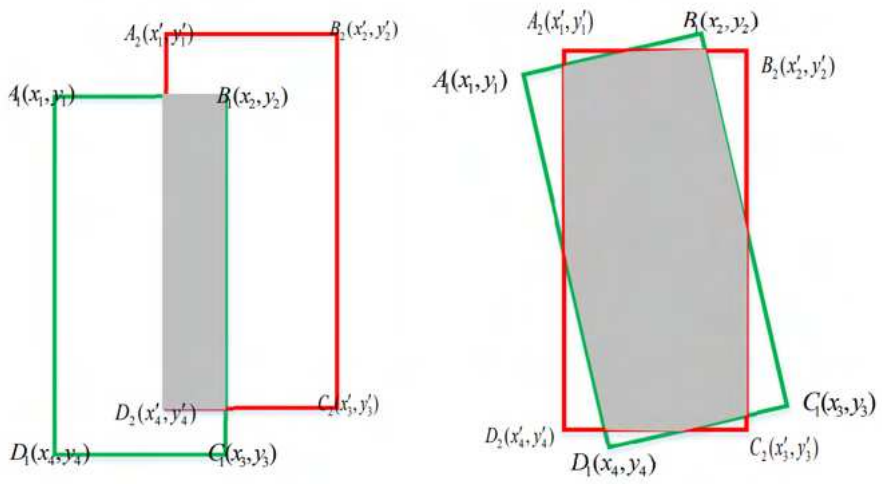
Essa métrica é uma boa alternativa quando FP e FN possuem a mesma importância na aplicação, pois ela avaliará o modelo nesses dois aspectos em um único score que variará de zero a um, sendo zero a pior pontuação e um a melhor.

A AP, ou Precisão Média, é calculada a área em baixo da curva do gráfico da Precisão e *Recall* e atribuído um score para avaliar o modelo considerando essas duas métricas (SZELISKI,

2022). Uma das diferenças do AP para Precisão, *Recall* e F1 *Score* é que essas métricas precisam da escolha de um limiar de confiança para ter uma comparação justa com outros modelos. Por conta disso, a AP é comumente utilizada como uma das métricas nos *benchmarks* de comparação de modelos de visão computacional por ser mais abrangente e precisa.

### *2.3.2 IoU*

Figura 6 – IoU - Em cinza área da interseção entre a *bounding box* prevista e a verdadeira




Fonte: Zhou *et al.* (2019).

Zhou *et al.* (2019) propuseram o IoU como uma métrica que avalia a união sobre a interseção das áreas das *bounding boxes* previstas pelo modelo comparando com as *bounding boxes* reais do objeto (ZHOU *et al.*, 2019). Na Figura 6 a parte em cinza representa visualmente a área do IoU, quanto maior essa área, mais próximo das coordenadas reais o modelo previu. A fórmula para obter o valor dessa métrica é definida da seguinte forma:

$$IoU(A,B) = \frac{A \cap B}{A \cup B} = \frac{A \cap B}{|A| + |B| - A \cap B} \tag{2.5}$$

Na Equação 2.5 (ZHOU *et al.*, 2019) a IoU é invariante à escala, portanto significa que a semelhança entre as duas formas comparadas A e B são independentes da escala do espaço em que estão inseridas. Isso foi essencial para popularizar o IoU como uma métrica de avaliação nas tarefas de Visão Computacional.

Nos comparativos dos modelos de Detecção de Objeto, é comum definir um *threshold* para considerar a *bounding box* prevista como “aceitável”. Isso permite uma personalização do

modelo de forma que se adéque ao rigor do desafio enfrentado.

### *2.3.3 mAP*

Segundo Henderson e Ferrari (2017), *Mean Average Precision (mAP)* utiliza o conceito do AP e IoU para calcular essa métrica em múltiplas classes de objetos. Essa métrica depende do *threshold* escolhido no IoU e é normalmente medida de duas formas:

- **mAP50** - É o mAP calculado com o *threshold* do IoU em 0,5. Essa métrica avalia a acurácia do modelo considerando as detecções mais fáceis; e
- **mAP50-95** - É o mAP calculado com o *threshold* do IoU variando de 0,5 a 0,95. Avaliando o modelo em diferentes níveis de dificuldade de detecção.

## 2.4 Conclusões Preliminares

Ante o exposto, o problema a ser tratado não é necessariamente novo, mas possui uma ênfase recente em trabalhos publicados na literatura técnico-científica. Uma vez que todos os conceitos fundamentais para o entendimento do trabalho, passaremos ao levantamento dessa bibliografia específica ao longo dos Capítulos 3 e 4.

# 3 TRABALHOS RELACIONADOS: *DATASETS*

Neste capítulo, apresentaremos *datasets* relacionados à classificação e segmentação de lixo. A escolha de um conjunto de dados adequado é crucial para o desenvolvimento de modelos de aprendizado de máquina capazes de detectar e segmentar diferentes tipos de resíduos. Serão analisadas as características de cada *dataset*, como tamanho, variedade de classes, características das imagens e finalidade do dataset. Além disso, discutiremos as principais limitações encontrados na utilização desses dados.

## 3.1 *Datasets* para Detecção de Lixo

O objetivo desta sessão é reunir e organizar os principais datasets públicos voltados à detecção de resíduos em imagens, disponibilizados pela literatura acadêmica. Esses datasets incluem imagens anotadas com *bounding boxes*. Esses *datasets* são insumos para os treinamentos dos modelos de *machine learning* capazes de determinar "o que" é o lixo e "onde" ele está, sem precisar de um contorno exato do objeto, fornecendo a posição e a dimensão aproximada do objeto detectado.

### *3.1.1 TACO-10*

O *dataset* proposto por Proença e Simões (2020) tem como um dos objetivos a detecção de lixo na natureza, podendo também conter imagens em ambiente e no contexto urbano. A base possui 1.500 imagens com 4.784 marcações. Suas imagens foram em sua maioria tiradas via *smart phones*, além de utilizarem um *crawler* no portal de hospedagem de imagens Flickr para encontrarem potenciais novas imagens de lixo. Nessa etapa de coleta, também utilizaram o site do projeto "Openlittermap" como fonte adicional de imagens.

Os autores criaram um site do projeto, em que qualquer pessoa pode subir novas imagens de lixo, como uma forma de permitir que todos possam colaborar com o aumento do *dataset*, essas imagens enviadas são avaliadas antes de entrarem permanentemente no *dataset* de imagens que precisam ser anotadas. Na etapa de anotação, Proença e Simões (2020) propôs uma nova ferramenta online no mesmo domínio onde é feito upload das imagens de lixo. Essa ferramenta faz exclusivamente marcação das imagens que ainda não possuem *labels* do *dataset* TACO. Para a tarefa de detecção do lixo, foi separado o *dataset* TACO-10 que possui as seguintes classes: *Bottle, Bottle cap, Can, Cigarette, Cup, Lid, Other, Plastic bag, Pop tab e Straw*.

Fonte: Proença e Simões (2020)
Figura 7 – Imagem do *dataset* TACO com lixo no ambiente urbano.

Fonte: Proença e Simões (2020)
Figura 8 – Imagem do *dataset* TACO com lixo na grama.

### *3.1.2 UAVVaste*

Segundo os autores de Kraft *et al.* (2021), o *dataset* foi composto por 772 imagens e 3.716 anotações de uma classe única representando o lixo. A motivação para a criação desse *dataset* foi a falta de dados específicos para o problema de detecção de lixo no ponto de vista aéreo de um drone assim como representado na Figura 13, diferenciando-se assim dos outros trabalhos na literatura como o TACO que possui a grande maioria das imagens tiradas do ponto de vista de um pedestre.

UAVVaste possui a característica de que, em suas imagens, o lixo normalmente é um objeto pequeno isolado no ambiente, este se alternando entre o urbano e o vegetativo.

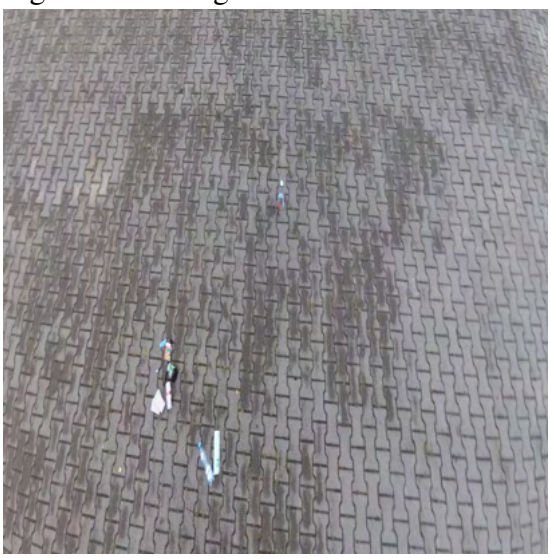
### *3.1.3 Wade-AI*

Conforme o estudo apresentado por Foundation (2016), o *dataset* Wade-AI inclui imagens capturadas do “Google Street View”. Essas imagens são bastante sortidas, possuindo um *background* variado, mas não necessariamente trazem algum objeto de lixo. O conjunto de dados é composto por 1.400 imagens com 2.200 marcações em uma única classe. Por conta da variabilidade da fonte da imagem, existe uma grande diversidade no ambiente e no tamanho dos

Figura 9 – Imagem do *dataset* UAVVaste.

Fonte: Kraft *et al.* (2021)

Figura 10 – Imagem do *dataset* UAVVaste.



Fonte: Kraft *et al.* (2021)

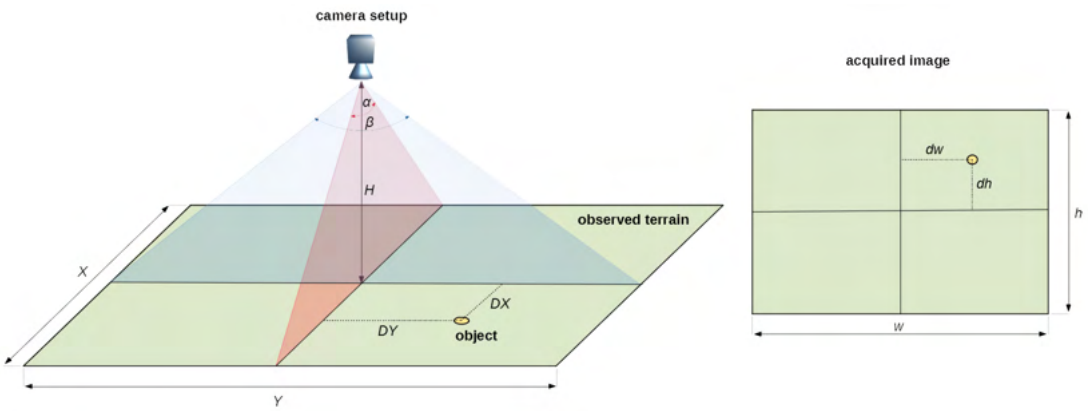
Figura 11 – Imagem do *dataset* UAVVaste.

Fonte: Kraft *et al.* (2021)

Figura 12 – Imagem do *dataset* UAVVaste.

Fonte: Kraft *et al.* (2021)

Figura 13 – UAVVaste - Representação do ângulo de captura de imagens contendo lixo via drone




Fonte: Kraft *et al.* (2021).

dados, como mostram da Figura 14 a Figura 17.


Fonte: Foundation (2016)
Figura 14 – Imagem do *dataset* Wade-AI.


Fonte: Foundation (2016)
Figura 15 – Imagem do *dataset* Wade-AI.

Fonte: Foundation (2016)
Figura 16 – Imagem do *dataset* Wade-AI.

Fonte: Foundation (2016)
Figura 17 – Imagem do *dataset* Wade-AI.

### *3.1.4 Trash-ICRA*

Trash-ICRA é um conjunto de dados proposto no trabalho do Fulton *et al.* (2020). Essa base de imagens possui figuras de lixo submersas em água e foi capturada do recorte dos *frames* dos vídeos gravados por um *Remotely Operated Vehicle (ROV)*. São 7.668 imagens no


Fonte: Fulton *et al.* (2020)

Figura 18 – Imagem do *dataset* Trash-ICRA.


Fonte: Fulton *et al.* (2020)

Figura 19 – Imagem do *dataset* Trash-ICRA.

tamanho de 480 por 360 pixels e 6.212 anotações, contendo 7 classes baseadas no material do objeto, são elas: lixo, objetos biológicos como plantas e animais e ROVs.

### *3.1.5 Trashnet*

Figura 20 – Trashnet - Imagens do dataset com o nome de sua respectiva classe abaixo.


Fonte: Yang e Thung (2016).

Proposto por Yang e Thung (2016), o *dataset* Trashnet é composto por 2.527 imagens, distribuídas da seguinte forma por classe:

- ***Glass*** - 501 imagens;
- ***paper*** - 594 imagens;

- ***cardboard*** - 403 imagens;
- ***plastic*** - 482 imagens;
- ***metal*** - 410 imagens; e
- ***trash*** - 137 imagens.

As fotos foram capturadas com *smartphones* Apple, modelos 7 Plus, 5S e SE, em ambientes com iluminação natural ou artificial, sempre com um fundo branco, o que garante que todas as imagens sejam bem iluminadas e possuam o mesmo cenário uniforme, como mostrado na Figura 20. Um aspecto que deve-se observar do *dataset* é a definição dos objetos classificados como lixo. Nas Figura 21, Figura 22 e Figura 23 por exemplo, são imagens da classe "trash"representada por objetos com pouca degradação ou uso, e essa abordagem de classificar esse tipo de objetos como lixo se repete ao longo de todo o conjunto de dados.

### *3.1.6 pLitterStreet*

O estudo elaborado por Mandhati *et al.* (2024) propõe um novo *dataset* com imagens de lixo em ambientes urbanos. A coleta foi realizada utilizando um veículo automotivo equipado de uma câmera GoPro Hero 9, a escolha dessa câmera se deu por conta dela possuir um GPS embutido (vide Figura 24), que foi útil no tralho já que os autores futuramente realizaram um mapeamento geoespacial com os dados coletados. Essa base de dados possui 13.064 imagens e 79.101 anotações, separadas em dois tipos de classificação, um com 4 classes e outro com 10 classes. As quatro classes são:

- *face mask*;
- *face maskpile*;
- *plastic*; e
- *trash bin*.

Por sua vez, as 10classes são:

- *bag*;
- *bottle*;
- *cup*;
- *face mask,*;
- *other plastic*;
- *pile*;
- *rope*;

Figura 21 – Imagem do *dataset* Wade-AI.

Fonte: Yang e Thung (2016)

Figura 22 – Imagem da classe *trash* da base de dados Trashnet.

Fonte: Yang e Thung (2016)

Figura 23 – Imagem do *dataset* Wade-AI.

Fonte: Yang e Thung (2016)

Figura 24 – pLitterStreet - Esquema de captura das imagens.


Fonte: Mandhati *et al.* (2024).

- *sachet*;
- *straw*; e

- *trash bin*.

### *3.1.7 PlastOPol*

Figura 25 – PlastOPol - Imagens do dataset.

Fonte: Córdova *et al.* (2022).

O conjunto de dados apresentado por Córdova *et al.* (2022) contém imagens de lixo em ambientes externos. A principal motivação para sua criação foi justamente a escassez de *datasets* voltados à coleta de imagens de lixo em cenários ao ar livre. Com uma única classe, o *datasets* inclui um total de 2.418 imagens e 5.300 anotações. A Figura 25 apresenta alguns exemplos das imagens desse conjunto de dados.

## 3.2 Datasets para Segmentação de Lixo

Assim como discutido na sessão anterior, o objetivo desta seção é reunir e organizar os principais *datasets* públicos voltados à segmentação de lixo, disponíveis na literatura acadêmica. Esses *datasets* contêm imagens anotadas com coordenadas que formam polígonos,

delimitando com precisão o contorno de cada objeto. Ao contrário da detecção, a segmentação oferece uma delimitação mais detalhada, gerando uma máscara que cobre exatamente a área ocupada pelo lixo.

### *3.2.1 TACO-1*

Figura 26 – TACO-1 - Recorte do dataset.


Fonte: Proença e Simões (2020).

O *dataset* TACO-1 também apresentado no trabalho do Proença e Simões (2020), possui um foco na detecção e segmentação do lixo. O que difere o TACO-1 e o TACO-10 é a marcação nas imagens e as *labels* usadas nas marcações, em que o TACO-1 é feito em formato poligonal para a segmentação semântica e o TACO-10 possui *bounding boxes* com as classes de cada objeto detectado.

Figura 27 – MJU-Waste - Recorte do dataset na etapa pós zoom.

Fonte: Wang *et al.* (2020).

### *3.2.2 MJU-Waste*

Proposto por Wang *et al.* (2020), o MJU-Waste é composto por 1.485 imagens para treino, 248 para validação e 742 para teste. As imagens foram capturadas pelos próprios autores utilizando objetos coletados no campus e registradas com o sensor Microsoft Kinect RGBD (HAN *et al.*, 2013). Esses objetos são considerados lixo potencial, seja porque poderiam ser descartados ou encontrados abandonados em vias públicas. Uma característica marcante das imagens é que os objetos são segurados pelos próprios autores, como mostrado na Figura 27, resultando na recorrente presença de mãos segurando os itens nas fotos com o mesmo *backgroud*.

### *3.2.3 TrashCan 1.0*

Focado em segmentação, o *dataset* TrashCan apresentado no trabalho de Hong *et al.* (2020) possui os mesmos métodos de captura de imagens mencionados na subseção 3.1.4 por isso suas imagens apresentam as mesmas características visuais. O que difere o dataset TrashCan 1.0 do Trash-ICRA, é o seu tamanho, resolução e a marcação. No tamanho, este possui 7.668 imagens e 6.706 marcações, a resolução dessas imagens é 480 por 270 pixels e as marcações foram feitas no nível de segmentação.

## 3.3 Conclusões Preliminares

A Tabela 1 possui um resumo geral dos *datasets* apresentados nesse capitulo com alguns comentários.

A existência de diversos tipos de *datasets* voltados para a detecção e segmentação semântica de lixo evidencia o crescente interesse da comunidade científica nesse tema. Nos trabalhos apresentados, observa-se uma ampla diversidade de ambientes, formatos, estados, tamanhos e ângulos em que o lixo foi capturado, refletindo a complexidade do problema. Cada

Tabela 1 – Principais *datasets* usados na literatura sobre resíduos sólidos.

| Nome | Autoria | Tipo | Quantidade | Comentário | Público? |
|---|---|---|---|---|---|
| TACO-10 | Proença e Simões (2020) | detecção | 1.500 | Lixo a céu aberto | sim |
| UAVVaste | Kraft *et al.* (2021) | detecção | 772 | Lixo a céu aberto, vista aérea (drone) | sim |
| Wade-AI | Foundation (2016) | detecção | 1.400 | Lixo a céu aberto | sim |
| Trash-ICRA | Fulton *et al.* (2020) | detecção | 7.668 | Imagens subaquáticas | sim |
| Trashnet | Yang e Thung (2016) | detecção | 2.527 | *Background* branco, classe lixo pouco deformada | sim |
| pLitterStreet | Mandhati *et al.* (2024) | detecção | 13.064 | Visão lateral de um veículo, *background* muito característico do local | sim |
| PlastOPol | Córdova *et al.* (2022) | detecção | 2.418 | Ambiente externo, imagens muito próximas do objeto | sim |
| TACO-1 | Proença e Simões (2020) | segmentação | 1.500 | Lixo a céu aberto | não |
| MJU-Waste | Wang *et al.* (2020) | segmentação | 1.485 | Ambiente fechado, objetos coletados no campus | sim |
| TrashCan 1.0 | Hong *et al.* (2020) | segmentação | 7.212 | Imagens subaquáticas | sim |

Fonte: elaborada pelo autor.

autor que publicou um *datasets* estava abordando uma necessidade específica, por exemplo, Trash-ICRA e TrashCan 1.0, o que ressalta a questão das classes utilizadas. Como não há uma convenção internacional para a classificação de lixo, a escolha de um *datasets* com determinadas classes e tipos de imagens pode ser mais eficaz dependendo da aplicação.

Os *datasets* TACO, UAVVaste, Wade-AI e PlastOPol possuem *backgrounds* diversos, mas, como mostrado nas Figuras Figura 7 e Figura 8, para o UAVVaste, as imagens frequentemente apresentam objetos muito pequenos ou indistinguíveis devido à distância, tamanho ou ângulo de captura. Esse fator também nos remete a outra questão relacionada ao *background*: embora seja possível identificar o material onde o objeto classificado como lixo está (grama, calçada, etc.), o contexto espacial da imagem torna-se subjetivo, dificultando uma interpretação mais clara do cenário.

O *dataset* Wade AI, assim como o TACO e o UAVVaste, apresenta uma grande diversidade de *backgrounds*. No entanto, ele não enfrenta o mesmo problema dos outros dois. O principal desafio aqui é a falta de consistência, devido à forma como o autor coletou as imagens de várias fontes, sem seguir um padrão de resolução ou critérios específicos para os locais onde o lixo foi capturado. Isso resulta em uma heterogeneidade nas imagens, o que pode impactar a qualidade e a uniformidade do *dataset*.

Para a proposta desse trabalho, não encontramos um *dataset* na literatura que atenda exatamente à realidade dos RSU que encontramos nas cidades do Brasil. Caracterizado por pontos de depósito de lixo nas calçadas ou meios fios, principalmente nos dias de coleta de lixo. Esses pontos geralmente possuem sacos plásticos coloridos amontoados envoltos em resíduos espalhados no chão. Essa diferença do cenário real de lixo urbano com os *datasets* públicos encontrados indica que esse campo ainda possui grande espaço para crescimento.

# 4 TRABALHOS RELACIONADOS: DETECÇÃO DE LIXO

Este capítulo apresenta uma revisão da literatura científica, visando fornecer um panorama das principais abordagens e resultados obtidos no campo da Visão Computacional envolvendo lixo. Serão discutidos os conjuntos de dados de lixo utilizados, as técnicas e seus resultados. A revisão será dividida em duas partes: a primeira tratará dos trabalhos com maior foco à detecção de objetos de lixo em imagens, e a segunda abordará a segmentação semântica do lixo. Ao final, serão apresentadas as conclusões preliminares e a aplicabilidade dos trabalhos no contexto da pesquisa aqui apresentada.

## 4.1 Métodos para Detecção de Lixo

A presente sessão tem como objetivo reunir alguns trabalhos que fazem uso de técnicas de visão computacional com o objetivo de detectar lixo.

### *4.1.1 Mandhati et al. (2024)*

O trabalho de Mandhati *et al.* (2024) tem como objetivo principal aplicar técnicas de *deep learning* para identificar lixo e pontos de descarte a partir de imagens capturadas ao nível da rua, utilizando uma câmera montada em um veículo. Durante a coleta, os autores mapearam os locais com presença de lixo, permitindo a criação de um mapa de calor de diversas cidades de Taiwan, que mostra a densidade de lixo na cidade. Além disso, o estudo propõe um novo *dataset* composto pelas imagens coletadas do veículo, conforme descrito anteriormente na subseção 3.1.6.

Os experimentos foram realizados em uma GPU Nvidia GeForce GTX 1080 Ti, com as imagens redimensionadas para 1024 x 1024 pixels. Os *frameworks* escolhidos para o treinamento foram: Faster R-CNN e RetinaNet, implementados no Detectron2 (WU *et al.*, 2019), além dos modelos YOLOv3 (JOCHER, 2020a) e YOLOv5 (JOCHER, 2020b).

A avaliação dos resultados obtidos se deu de três formas, treinando os quatro *frameworks* escolhidos no *dataset* pLitterStreet com quatro classes ( *plastic, pile, facemask, trash-bin*), com dez classes (*bag, bottle, cup, facemask, other plastic, pile, rope, sachet, straw, trash-bin*) e avaliando os resultados nos outros *datasets* (2). Para este último, todas as classes foram remapeadas para uma única classe e os resultados apresentados são da métrica AP.

Tabela 2 – Avaliação em outros *datasets* remapeados para uma classe.

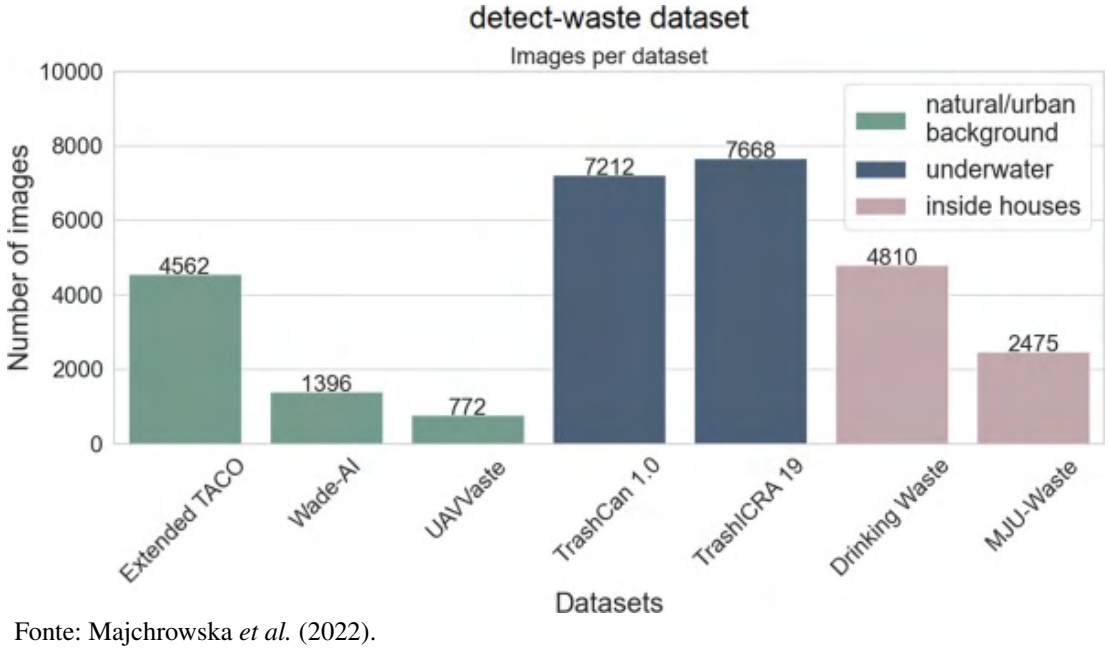
| | PlastOPol | TACO | pLitterStreet |
|---|---|---|---|
| Faster R-CNN + treinado no PlastOPol | 0,638 | 0,369 | 0,076 |
| YOLO-v5l + treinado no PlastOPol | 0,641 | 0,314 | 0,081 |
| Faster R-CNN + treinado no TACO | 0,551 | 0,436 | 0,079 |
| YOLO-v5l + treinado no TACO | 0,550 | 0,454 | 0,125 |
| Faster R-CNN + treinado no pLitterStreet | 0,259 | 0,244 | 0,409 |
| YOLO-v5l + treinado no pLitterStreet | 0,257 | 0,321 | **0,440** |

Fonte: Adaptado de Mandhati *et al.* (2024).

### *4.1.2 Majchrowska et al. (2022)*

Elaborado por (MAJCHROWSKA *et al.*, 2022), esse trabalho propõe um novo conjunto de dados para detectar e classificar resíduos, que é nada mais do que uma coleção dos outros *datasets* (Extended TACO, Wade-Ai, UAVVaste, TrashCan 1.0, TrashICRA, Drinking Waste e MJU-Waste) disponíveis publicamente. Os autores não informaram exatamente o tamanho final do *dataset*, mas afirmaram que possui mais de 28.000 imagens e 40.000 marcações. As marcações de cada base foram unificadas nas seguintes categorias: *bio, glass, metal and plastic, non-recyclable, other, paper, and unknown*.

Figura 28 – Detect-waste - Contagem de imagens por *dataset* e classificação dos tipos de *dataset*.




Fonte: Majchrowska *et al.* (2022).

Os autores também apresentam um detector de duas etapas uma que faz a detecção e

a outra a classificação do lixo. O EfficientDet-D2 é usado para detectar o lixo e o EfficientNet-B2 para classificar os resíduos detectados nas sete categorias propostas. O classificador foi treinado de forma semi-supervisionada utilizando imagens não rotuladas. Essa abordagem atingiu até 0,7 de AP na detecção de resíduos e cerca de 0,75 de precisão de classificação no conjunto de dados de teste.

Infelizmente, a reprodutibilidade dos resultados foi prejudicada porque o autor não disponibilizou o conjunto exato de dados, nem o *split* utilizado. No repositório do projeto, foram fornecidos links para o download individual de cada base. Além disso, havia outros *datasets* não mencionados inicialmente no estudo, e um dos links estava indisponível ou desatualizado. Esses fatores dificultaram a comparação entre os resultados e comprometeram a reprodutibilidade.

### *4.1.3 Kraft et al. (2021)*

O estudo feito por Kraft *et al.* (2021), anteriormente abordado na subseção 3.1.2 pelo dataset proposto, tem como objetivo a utilização de *drones* para patrulhar uma região, esses *drones* farão a detecção de lixo na área e o local em que foi encontrado o resíduo é adicionado em um mapa. O *drone* estará equipado com uma câmera que fará a detecção do lixo utilizando modelos de *deep learning* durante a sua volta automática da patrulha. Para os treinamentos foi utilizado variantes do Yolov4, Yolov3 e EfficientDet e o MobileNetV2 SSD e o dataset proposto pelo trabalho o UAVVaste. Os principais resultados obtidos estão na Tabela 3, para a coluna do *recall* foi calculado a média dos valores de recall de cada modelo para objetos pequenos, médios e grandes, com objetivo de facilitar comparações futuras. Os autores também levaram em consideração o tempo de inferência portanto optaram pelo modelo do Yolov4 com a metade da precisão para balancear o *trade-off* da precisão de velocidade de inferência no hardware limitado do *drone*.

O estudo realizado por Kraft *et al.* (2021), abordado anteriormente na subseção 3.1.2 pelo *datasets* proposto, tem como objetivo o uso de *drones* para patrulhar uma região, detectando resíduos e mapeando os locais onde são encontrados. O *drone* estará equipado com uma câmera que realizará a detecção utilizando modelos de*deep learning* durante seu percurso de patrulha automática. Foram utilizados, para os treinamentos, variantes dos modelos Yolov4, Yolov3, EfficientDet e MobileNetV2 SSD, além do *datasets* UAVVaste, proposto no trabalho. Os principais resultados estão apresentados na Tabela 3. Para a coluna de *recall*, foi calculada a média dos valores de *recall* para objetos pequenos, médios e grandes, a fim de facilitar

comparações futuras. Os autores também consideraram o tempo de inferência e, por isso, optaram pelo modelo Yolov4 com precisão reduzida pela metade, buscando equilibrar o *trade-off* entre a precisão e a velocidade de inferência no hardware limitado do *drone*.

Tabela 3 – Avaliação do UAVVaste em diferentes modelos.

| | mAP50 | mAP50-95 | R |
|---|---|---|---|
| YOLOv4 | **0,785** | 0,476 | 0,521 |
| YOLOv3 | 0,694 | 0,342 | 0,363 |
| YOLOv4-CSP | 0,736 | 0,424 | 0,475 |
| YOLOv4-tiny-3l | 0,615 | 0,336 | 0,350 |
| YOLOv4-tiny | 0,566 | 0,280 | 0,241 |
| YOLOv3-tiny | 0,239 | 0,064 | 0,069 |
| EfficientDet-d1 | 0,669 | 0,338 | 0,474 |
| EfficientDet-d3 | 0,751 | 0,440 | 0,543 |
| MobileNetV2 SSD | 0,545 | 0,255 | 0,385 |

Fonte: Kraft *et al.* (2021)(Adaptado)

## 4.2 Conclusões Preliminares

Os trabalhos apresentados neste capítulo foram selecionados por terem maior correlação com o objetivo deste estudo, ou, de forma mais ampla, com o tema da detecção de lixo em áreas urbanas externas, além da possibilidade de comparar as métricas obtidas nos mesmos conjuntos de dados.

O método proposto na subseção 4.1.1 utiliza o *dataset* mencionado na subseção 3.1.6, caracterizado por imagens urbanas das cidades de Taiwan, com *backgrounds* muito similares. As imagens de lixo focam no meio-fio, destacando o objeto, o que facilita sua detecção. Na última linha da Tabela 3, onde os autores alcançam o melhor resultado em seu próprio *dataset*, comparado com os demais, percebe-se que a semelhança das imagens do pLitterStreet com as do TACO e PlastOPol contribuiu para que o modelo obtivesse resultados consistentes em todas as bases, com variação de 0,20 a 0,50 na precisão, ilustrando a complexidade da tarefa de detecção de lixo.

O trabalho descrito na subseção 4.1.2 apresenta uma excelente iniciativa ao tentar padronizar as classes de lixo urbano utilizando *datasets* públicos. No entanto, os autores não disponibilizam a base de dados coletada para os treinos e validações, apenas fornecem links em seu repositório para obtenção dos *datasets* selecionados. Essa abordagem inviabiliza uma

comparação justa entre o *dataset* utilizado na etapa de treino e validação e os resultados deste trabalho.

O estudo de Kraft *et al.* (2021) atingiu com sucesso o objetivo proposto, apresentando resultados sólidos, como um mAP50 de 0.785. No entanto, as limitações do estudo estão relacionadas ao contexto das imagens capturadas por *drones* em áreas amplas e remotas, onde pequenos resíduos estão dispersos no solo. Esse cenário pode não refletir adequadamente a realidade de outros ambientes, limitando a aplicabilidade dos resultados a situações similares. Além disso, o estudo não representa de forma precisa a realidade dos RSU encontrados em grandes metrópoles brasileiras, onde o acúmulo de lixo ocorre em contextos urbanos mais complexos e densamente povoados.

Por outro lado, o presente trabalho foca na detecção de lixo em áreas urbanas, como grandes cidades, utilizando câmeras de segurança e monitoramento de vias públicas, onde o lixo se acumula frequentemente em calçadas e meios-fios. Esse ambiente urbano aumenta significativamente a complexidade do problema, pois a variabilidade dos cenários, combinada com o tipo e o tamanho dos resíduos, afeta diretamente a acurácia dos modelos de detecção. A diversidade de condições nesses espaços exige soluções mais robustas e adaptáveis, em comparação com aquelas desenvolvidas para ambientes mais controlados e homogêneos.

# 5 METODOLOGIA

Neste capítulo, serão apresentados o conceito de lixo adotado neste estudo, uma descrição detalhada dos dados utilizados para o treinamento e avaliação, além de uma discussão sobre as classes escolhidas e o tipo de anotação utilizado nos *frameworks* aplicados no desenvolvimento do modelo de *deep learning*.

## 5.1 Definição de RSU (Complementar)

Os RSU se caracterizam pela diversidade em tamanho, cores, formas, densidade e tipo de material. Isso pode ser observado na classificação realizada pela prefeitura de Fortaleza para os RSU coletados em 2023, que inclui as seguintes classes: CAPINA, CHORUME, CLASSE IIA COMUM, CLASSE IIA UMIDADE > 20%, COLETA SELETIVA, COMPOSTAGEM - LODO DE ETE, DOMICILIAR, ENTULHO, LIXO ESPECIAL URBANO, OUTROS, PODA TRITURADA, PODAÇÃO, RCC - CLASSE A, RCC - CLASSE B, RESÍDUOS INERTES, TRANSBORDO, TRONCO DE PODA e VARRIÇÃO.

Essa classificação da ACFOR (2024) foi realizada com base na NBR 10.004/04 (ABNT, 2004), dela se se destacam os itens CAPINA representando um total de 5,91% do volume total dos resíduos em 2023, DOMICILIAR com 36,61%, ENTULHO 17,88%, LIXO ESPECIAL URBANA 23,38%, PODAÇÃO 2,44% e VARRIÇÃO 1,98 % pelo volume de resíduos dessa categoria coletados no ano de 2023. Com isso, um sistema que cobrisse essas classes seria capaz de atender a 88,23 % da demanda de RSU da cidade de Fortaleza.

## 5.2 Conjunto de Dados

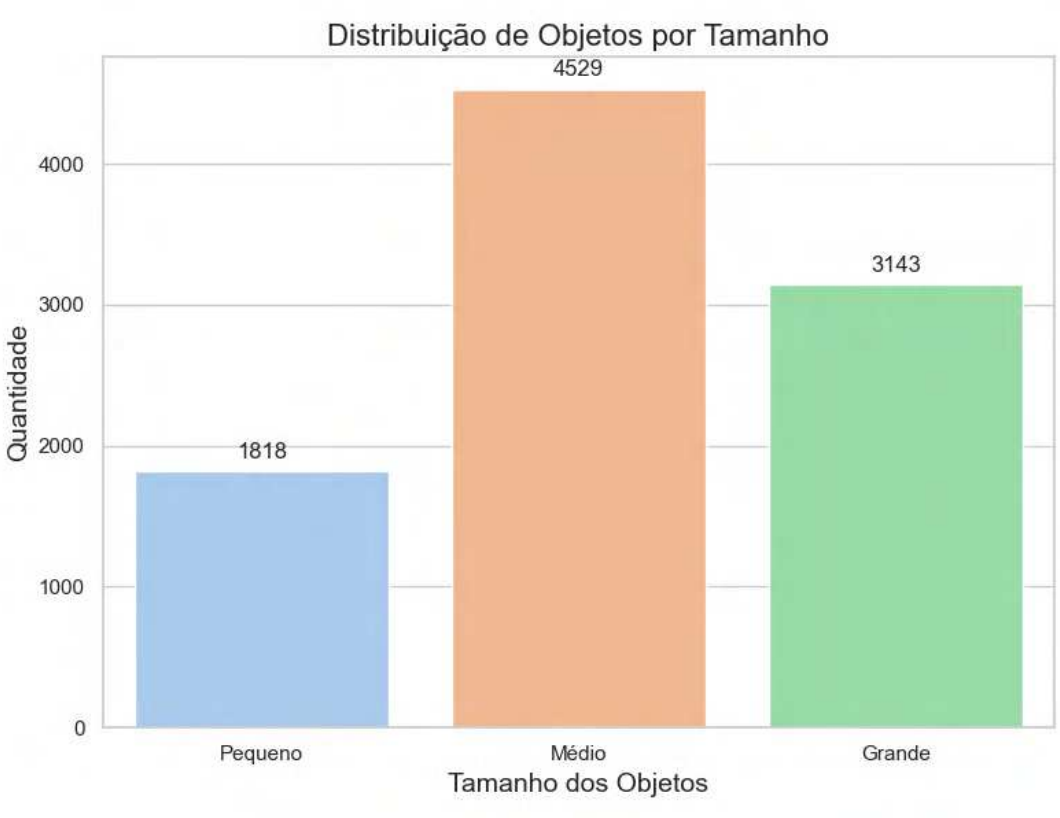
A coleta de dados foi realizada por três métodos distintos: imagens capturadas em primeira pessoa, utilizando *smartphones* tanto na orientação horizontal quanto vertical; fotos e vídeos registrados em um carro, com a câmera do *smartphone* apontada para a rua através das janelas, e, posteriormente, extraindo *frames* dos vídeos para transformá-los em imagens da base de dados; além disso, também foram utilizadas câmeras de segurança do campus do PICI. Adicionalmente, um *dataset* privado, contendo imagens de câmeras de vigilância e vídeos, foi incorporado. Nesse estudo, esse conjunto de dados será nomeado como Dataset A. Essa diversidade na coleta de dados oferece uma ampla variedade de ângulos e *backgrounds*, com o intuito de construir um modelo mais robusto e abrangente. A versão final do *dataset* é composto

de 9.231 imagens e 9.490 marcações. É possível conferir a distribuição do tamanho dos objetos marcados seguindo os critérios do Microsoft COCO Challenge Dataset (2022)(Tabela 4), na Figura 29 .

Tabela 4 – Definição dos tamanhos de objetos no MS COCO

| **Tamanho** | **Área mínima (px)** | **Área máxima (px)** |
|---|---|---|
| Pequeno | $0 \times 0$ | $32 \times 32$ |
| Médio | $32 \times 32$ | $96 \times 96$ |
| Grande | $96 \times 96$ | $\infty \times \infty$ |

Figura 29 – Dataset-A - Distribuição de objetos por tamanho.




Fonte: Elaborado pelo autor.

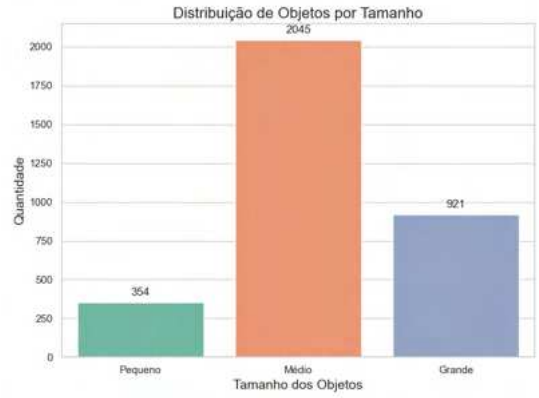
Para auxiliar na montagem e organização do Dataset A, foram criados diversos *scripts* em Python. Esses *scripts* foram responsáveis por tarefas como a extração de *frames* de vídeos, organização dos arquivos ".txt" nas pastas conforme os requisitos do YOLOV8, divisão do conjunto de dados em treino, teste e validação, renomeação de imagens com caracteres especiais que poderiam causar problemas, entre outras atividades essenciais para o processo.

Nas anotações das imagens foi levado em consideração a classificação de RSU vista na seção 5.1 para definir o que seria considerado lixo e marcado na rua, principalmente as classes

com maior volumetria encontradas, nesse estudo todas elas foram consideradas como uma única classe representando lixo.

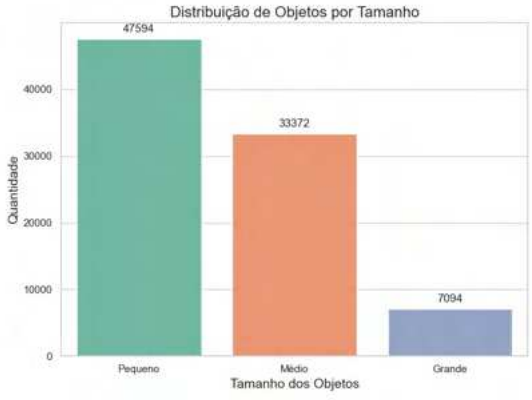
Para fins de comparação com Dataset A, as figuras, Figura 30 e Figura 31 permitem observar a contagem do tamanho das *bounding boxes* dos lixos. Além disso, foi possível encontrar na internet base com 732 imagens e 3.320 marcações e 15.138 imagens e 88.060 marcações de objetos de lixo, para o UAVVaste e pLitterStreet respectivamente.

Figura 30 – UAVVaste - Distribuição de objetos por tamanho.




Fonte: elaborado pelo autor.

Figura 31 – pLitterStreet - Distribuição de objetos por tamanho.




Fonte: elaborado pelo autor.

## 5.3 Frameworks utilizados

Para a tarefa de detecção de lixo, optou-se pelo YOLOV8 (JOCHER *et al.*, 2023), devido ao seu desempenho superior em relação às versões anteriores da família YOLO, oferecendo resultados mais eficazes, precisos e estáveis. Outro fator decisivo foi a facilidade de instalação e uso, juntamente com o suporte de uma documentação ampla e clara, além de contar com uma comunidade ativa que facilita a resolução de problemas.

O treinamento foi realizado algumas vezes, à medida que a quantidade de imagens da base de dados aumentava, novas rodadas de treinamento eram realizadas. Também foram realizados testes nas diferentes versões do YOLOV8, como YOLOV8-l, YOLOV8-m e YOLOV8-x. Além de ajuste em diversos parâmetros nos treinos com a finalidade de obter melhores resultados.

Para a segmentação, será utilizado o modelo SAM2, considerado o estado da arte no campo da segmentação em Visão Computacional (RAVI *et al.*, 2024). O SAM2 oferece um desempenho generalista excepcional, adaptando-se a diversos cenários, o que é crucial no

contexto da detecção de RSU, dada a sua variabilidade. Além disso, a possibilidade de integração do SAM com o YOLOV8 foi um fator determinante, já que ele aceita *prompts* como máscaras, pontos e *bounding boxes*, otimizando o processo de segmentação em conjunto com a detecção.

## 5.4 *Script* de processamento de imagens

Nesse trabalho, estamos propondo um *script* de processamento de imagens que possui duas etapas. Na primeira parte, a imagem de entrada passa pelo modelo YOLOV8, que detectará objetos de lixo na figura, marcando a característica *bounding box* em seu entorno. Essa *bounding box* servirá como *prompt* de entrada para a segunda etapa do *script*, no qual o SAM fará a segmentação da imagem, o que se espera aqui é que apenas os objetos considerados como lixo sejam segmentados. Então o *script* finaliza exportando dois tipos de imagens, uma igual à imagem de entrada, mas com a *bounding box* sobreposta e a área segmentada dentro da *bounding box* pintada com alguma cor, a segunda imagem exportada é uma máscara da imagem com o objeto detectado como lixo isolado do restante da figura e com essa máscara, é calculado a área ocupada por lixo na imagem em relação ao total de *pixels* da imagem de entrada.

## 5.5 Validação dos resultados

Os resultados obtidos serão avaliados da seguinte forma: todos os modelos serão treinados e avaliados no mesmo ambiente, GPU NVIDIA GeForce RTX 2070 Max-Q, com os mesmos *datasets*, o Dataset-A 5.2, pLitterStreet 3.1.6 e UAVVaste 3.1.2. Então, cada modelo após treinamento será utilizado para processar um conjunto de testes de imagens que fará a detecção e segmentação mencionado na seção 5.4.

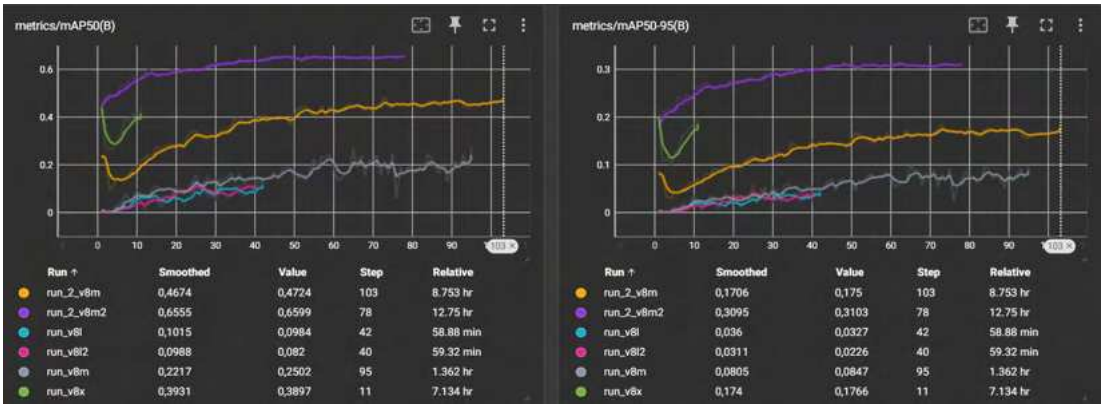
# 6 RESULTADOS

Este capítulo apresenta e discute os resultados obtidos após diversas rodadas de treinamento com o Dataset A 5.2 apresentado no capítulo anterior. Comparações com os resultados obtidos pelos trabalhos relacionados serão feitas utilizando os *datasets* públicos conhecidos disponíveis.

## 6.1 Modelo proposto

Com o YOLOV8m obtivemos os melhores resultados para o Dataset A. Na melhor rodada de treino, obtivemos um *score* de 0,659 para o mAP e 0,31 para mAP50-95, para Precisão e Recall o modelo pontuou 0,73 e 0,59 respectivamente. Nas Figuras 32 e 33, é possível observar esses resultados em comparação com algumas das outras rodadas de treino, o melhor resultado possui o nome **run_2_v8m2** representado pela linha roxa nos gráficos, que chamaremos de Modelo A. Para a segmentação, estaremos utilizando o modelo **sam2_hiera_base_plus** em todos os teste.

Figura 32 – Modelo A - mAP50 e mAP50-95.




Fonte: elaborado pelo autor.

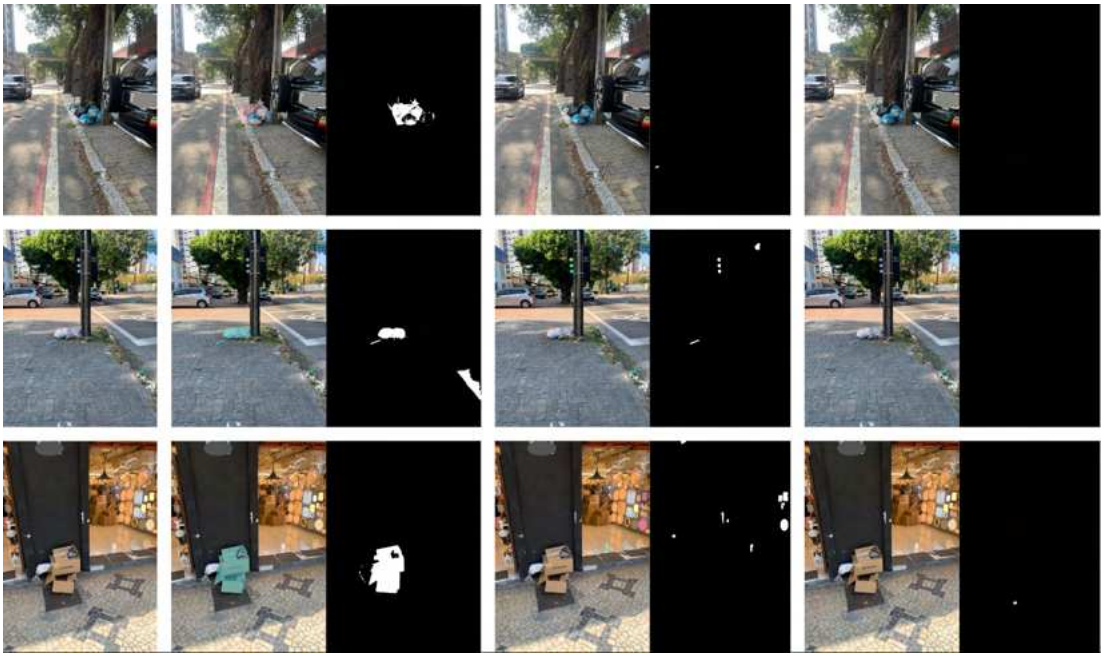
Os mosaicos das Figuras 34, 35 e 36 ilustram as etapas do processo de teste. Na primeira coluna está a imagem de entrada, e para as outras três teremos duas imagens por coluna, na qual a imagem da esquerda mostra o resultado da atividade de detecção de lixo na imagem de entrada, representado pela *bounding box* verde retangular e dentro dessa área estará a segmentação do objeto detectado pelo modelo como lixo. O modelo representante dessas três colunas da esquerda para a direita é Modelo A, UAVVaste e pLitterStreet.

Ao analisar o desempenho geral dos três modelos nos mosaicos mencionados, como

Figura 33 – Modelo A - Precisão e Recall.

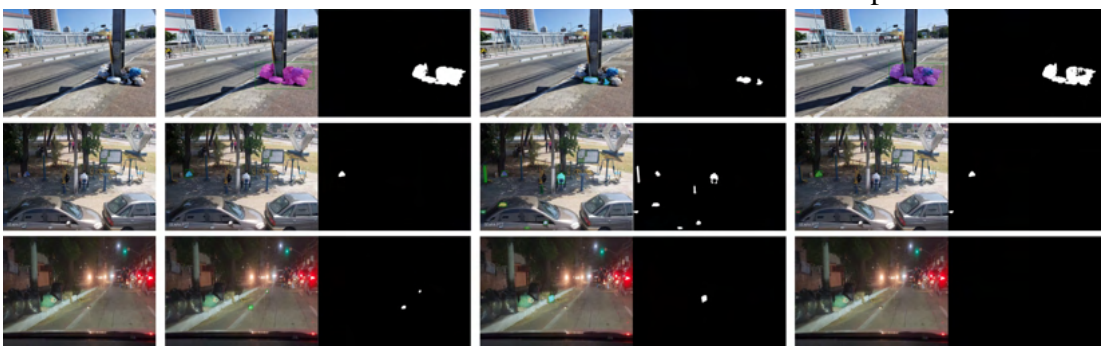

Fonte: elaborado pelo autor.

Figura 34 – Imagem de entrada (à esquerda) e os resultados em cada modelo
Entrada Modelo A UAVVaste pLitterStreet.



Fonte: elaborado pelo autor.

Figura 35 – Imagem de entrada (à esquerda) e os resultados em cada modelo
- Modelo A UAVVaste pLitterStreet.



Fonte: elaborado pelo autor.

Figura 36 – Imagem de entrada (à esquerda) e os resultados em cada modelo - Modelo A UAVVaste pLitterStreet.

Fonte: elaborado pelo autor.

esperado, o Modelo A se destacou nas imagens de teste do Dataset A, superando os outros modelos treinados em seus respectivos *datasets*. No entanto, ainda podemos observar algumas limitações na detecção feita pelo Modelo A. Na segunda linha da Figura 35 há uma imagem capturada por uma câmera de segurança. Nesse cenário, o modelo conseguiu detectar três sacos de lixo, mas não identificou uma caixa ao lado deles. Isso ocorreu devido ao formato da caixa, com ângulos retos, algo incomum nas imagens do Dataset A, além da distância do objeto. Como mostrado na Figura 29, a maior parte das marcações de lixo no Dataset A se concentra em tamanhos médio e grande, o que influenciou negativamente a confiança do modelo em classificar a caixa como lixo.

Ainda na Figura 35, na última linha, encontramos o pior resultado desse teste para o Modelo A. A imagem foi tirada de dentro de um veículo em movimento durante a noite, apresentando um cenário totalmente novo. Nessa situação, a aplicação não demonstrou robustez suficiente para identificar nenhum dos RSU presentes. Em vez disso, ele identificou incorretamente um objeto que fazia parte do plano de fundo (*background*).

Em relação aos resultados dos outros modelos, de modo geral, eles não se saíram tão bem nas imagens de teste do Dataset A. O melhor desempenho do UAVVaste foi observado nas imagens capturadas por câmeras de segurança. Nessas imagens, devido à distância, os objetos aparecem perspectivamente menores, um cenário mais familiar para esse modelo, como mostrado na Figura 30.No entanto, essa tendência de identificar objetos menores como lixo fez com que o UAVVaste cometesse diversos erros, classificando incorretamente vários elementos do *background* como lixo, tanto nessa quanto em outras fotos do teste. O pLitterStreet apresentou um desempenho fraco, detectando objetos em apenas três das imagens de teste. Dessas quatro detecções realizadas, apenas duas realmente correspondiam a resíduos de lixo, evidenciando as limitações do modelo nesse contexto.

Figura 37 – Modelo A - Teste com níveis de ruído aditivo.

Fonte: elaborado pelo autor.

Figura 38 – UAVVaste - Teste com níveis de ruído aditivo.

Fonte: elaborado pelo autor.

Figura 39 – pLitterStreet - Teste com níveis de ruído aditivo.

Fonte: elaborado pelo autor.

Utilizamos a técnica de ruído Gaussiano (FOI *et al.*, 2008) para simular adversidades e testar a robustez dos modelos, aumentando gradativamente a intensidade do ruído em cada imagem, de 20 em 20% até um total de 100%. Para esse teste, selecionamos uma imagem com boa iluminação e um nível de dificuldade baixo para a detecção de lixo. As Figuras 37, 38 e 39 mostram os resultados obtidos. O Modelo A e o pLitterStreet conseguiram realizar detecções em imagens com baixa intensidade de ruído, mas o modelo UAVVaste não obteve nenhum acerto, mesmo em condições de ruído reduzido.

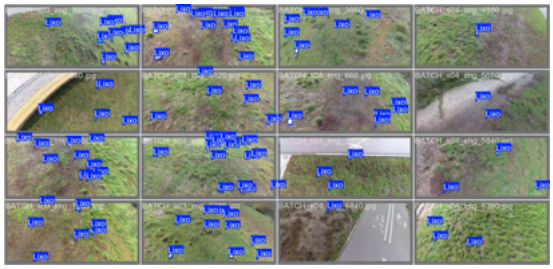
Tabela 5 – Avaliação geral dos modelos por dataset.

| **Modelo** | **UAVVaste** | | | **pLitterStreet** | | | **DatasetA** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **R** | **mAP50** | **mAP50-95** | **R** | **mAP50** | **mAP50-95** | **R** | **mAP50** | **mAP50-95** |
| YOLOv5l (2) | - | - | - | - | 0,4400 | - | - | - | - |
| YOLOv4 (3) | 0,5210 | **0,7850** | **0,4700** | - | - | - | - | - | - |
| YOLOv8m (Modelo A) | 0,0373 | 0,0076 | 0,0015 | 0,0600 | 0,0170 | 0,0036 | **0,5900** | **0,6590** | **0,3150** |
| YOLOv8m (UAVVaste) | **0,6510** | 0,7080 | 0,4050 | 0,2370 | 0,1660 | 0,0828 | 0,0113 | 0,0010 | 0,0003 |
| YOLOv8m (pLitterStreet) | 0,6060 | 0,5720 | 0,3080 | **0,5740** | **0,6350** | **0,3680** | 0,1460 | 0,0816 | 0,0307 |

Fonte: Elaborado pelo autor.

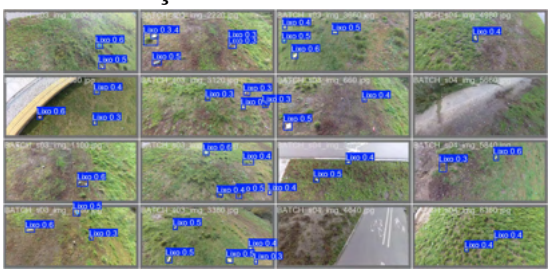
A Tabela 5 é constituída das principais métricas coletadas no processo de teste. Apesar da tarefa difícil, o Modelo A se saiu relativamente bem com mAP50 em 0,658. Mas não consegue realizar o mesmo feito nos outros *datasets* pela diferença nos cenários e da estratégia de marcação escolhida por cada trabalho. Apesar dos scores baixos, é notável como o Modelo A conseguiu detectar alguns resultados interessantes nas Figuras 41 e 43.

Figura 40 – Validação Model A - Marcações (*labels*) do dataset UAVVaste.



Fonte: Elaborado pelo autor.

Figura 41 – Validação Model A - detecções no dataset UAVVaste.

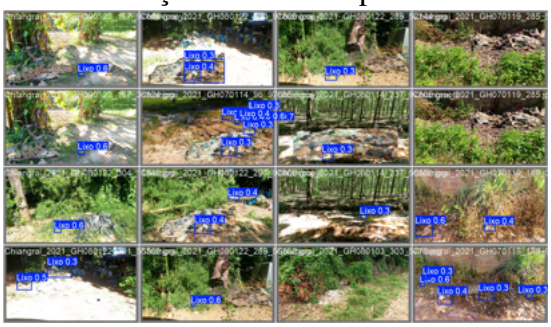


Fonte: Elaborado pelo autor.

Figura 42 – Validação Model A - Marcações (*labels*) do dataset pLitterStreet.

Fonte: Elaborado pelo autor.

Figura 43 – Validação Model A - detecções no dataset pLitterStreet.



Fonte: Elaborado pelo autor.

## 6.2 Limitações e Ameaças à Validade

O presente trabalho possui algumas limitações e vieses que podem ter influenciado os resultados obtidos. Primeiramente, a busca bibliográfica foi realizada predominantemente em inglês e português, o que pode ter excluído pesquisas relevantes publicadas em outros idiomas, especialmente aquelas com foco regional ou cultural. Além disso, foram priorizados trabalhos que apresentavam resultados quantitativos com métricas comparáveis às geradas pelo modelo YOLOv8, o que restringiu o escopo da comparação. Isso limitou a abrangência dos estudos analisados, excluindo aqueles que poderiam trazer perspectivas diferentes.

Outro ponto que merece destaque é a escolha dos modelos comparados. Optei por restringir a análise a modelos viáveis em termos de reprodutibilidade, considerando aqueles que utilizavam métricas semelhantes e cujos dados fossem abertos e já marcados, com indicação do *split* realizado. Isso impôs um filtro adicional, excluindo trabalhos cujos dados não estavam disponíveis publicamente ou que apresentavam diferenças significativas em seu escopo, como detecção de lixo em ambientes fechados ou em contextos controlados.

Uma limitação importante foram os *frameworks* utilizados. Decidi utilizar o YOLOv8

para todos os modelos, mas poderia ter testado tecnologias mais atuais que poderiam entregar resultados melhores como Mask R-CNN (HE *et al.*, 2018) e SAHI (AKYON *et al.*, 2022). O fato de utilizar YOLOv8 para treinamento em todos os modelos, apesar de não ser o *framework* utilizado pelo estudo original pode ter influenciado diretamente os resultados, visto que cada *frameworks* pode ter suas próprias características e desempenhos específicos. Além disso, o trabalho focou nas métricas de mAP, Recall e Precisão, sem avaliar a velocidade de detecção e segmentação, crucial para a aplicação desses sistemas em tempo real. A falta dessa análise acaba limitando o entendimento completo do desempenho dos modelos, especialmente em cenários práticos, onde a eficiência computacional é um fator determinante.

Em relação ao treinamento dos modelos, apenas o Modelo A, que era o foco principal deste estudo, passou por várias rodadas de *fine-tuning*. Já os modelos UAVVaste e pLitterStreet foram treinados apenas uma vez, utilizando os parâmetros sugeridos automaticamente pelo YOLOv8m. Um cuidado maior no ajuste fino desses modelos poderia ter proporcionado resultados mais robustos e comparáveis.

Por fim, este trabalho não incluiu o uso de técnicas tradicionais de visão computacional, como Transformada de Fourier (NETO, 1999), *Random Sample Consensus* (RANSAC) (CANTZLER, 1981), Detecção de Arestas (SORIA *et al.*, 2023), Detecção de *Outliers* com Segmentação (BEVANDIĆ *et al.*, 2019), que poderiam ter servido como uma base de comparação para o desempenho das abordagens mais modernas, como o YOLOv8m e o SAM. Essa ausência pode ser vista como uma limitação, já que a inclusão dessas técnicas permitiria uma avaliação mais completa do progresso alcançado pelos métodos contemporâneos.

# 7 CONSIDERAÇÕES FINAIS

Nesse estudo, foi desenvolvido e avaliado um modelo de *Deep Learning* capaz de detectar diferentes tipos de resíduos sólidos em áreas urbanas. A parte da segmentação resulta da integração do Modelo A com o *prompt* que é passado para o modelo SAM 2.

Também fez parte deste trabalho uma revisão dos *datasets* disponíveis publicamente na literatura para detecção e segmentação de lixo, além de um levantamento das técnicas que esses estudos têm aplicado para os respectivos problemas.

Foi realizada uma coleta e tratamento de dados, além de uma marcação intensiva nas imagens do Dataset A, o qual foi amplamente utilizado no presente estudo assumindo que este *dataset* mais se aproxima da realidade que encontramos nas ruas das grandes metrópoles do Brasil. Além disso, cada um dos modelos desenvolvidos com os *datasets* dos trabalhos relacionados foram avaliados experimentalmente. Tais experimentos levaram a etapa de avaliação utilizando métricas entendidas como essenciais para fins de comparação com outros trabalhos.

Muito embora os resultado do modelo proposto não apresentem um desempenho ideal no Dataset A, esse cenário nos permitiu verificar a viabilidade de um modelo de detecção de classe única para lixo e com assertividade compatível com o estado da arte. No entanto, ficou claro que uma abordagem mais eficaz para o problema seria a utilização de diversas classes de resíduos, o que facilitaria a detecção de objetos com características semelhantes, como também permitiria uma classificação mais precisa e alinhada àquela aplicada por prefeituras no gerenciamento dos Resíduos Sólidos Urbanos.

Como comentado na seção 6.2, não fizemos avaliação levando em consideração o tempo de processamento, portanto a aplicabilidade desses modelos apresentados em infraestruturas reais de computação será deixada para trabalhos futuros. Uma futura segunda fase deste trabalho poderia focar em modelos de classificação mais avançados e versáteis, tais como o Mask R-CNN ou as encarnações mais recentes do YOLO (e.g., v9 ou v10), para fins de comparação de resultados. Além disso, técnicas como o *Slicing Aided Hyper Inference* (SAHI), desenvolvida para melhorar a detecção de objetos pequenos em imagens de alta resolução, poderiam ser incorporadas para aumentar a eficiência e precisão da detecção em imagens mais amplas.

Outro passo importante seria o treinamento de modelos de segmentação semântica, o que potencialmente aumentaria a acurácia, principalmente ao lidar com a complexidade de diferentes tipos de resíduos. A segmentação de múltiplas classes de lixo, como lixo ensacado, poda e entulho, reduziria a discrepância entre objetos de uma mesma classe, melhorando o

desempenho geral do modelo. É importante salientar que subdividir os *datasets* para usos personalizados, como diferentes tipos de resíduos, também traria benefícios na adaptação dos modelos às características específicas de cada caso.

Ademais, uma abordagem em duas etapas, similar ao que foi visto em trabalhos anteriores, poderia ser implementada: uma primeira fase de detecção de objetos seguida de uma segunda fase de segmentação semântica mais detalhada. A inclusão da detecção de eventos, como identificar quando uma pessoa deposita lixo em local inadequado, poderia ampliar o escopo do sistema, tornando-o ainda mais relevante no combate ao descarte irregular.

Por fim, o uso de técnicas de *data augmentation* seria fundamental para aumentar o volume de imagens com lixo em diferentes cenários, ampliando assim a robustez dos modelos. Esse conjunto de melhorias abre novas possibilidades para o desenvolvimento de soluções mais robustas e aplicáveis ao gerenciamento de lixo urbano, tanto em termos de detecção quanto de segmentação, alinhadas às necessidades reais das cidades.

# REFERÊNCIAS

ACFOR. **Resíduos Sólidos Fortaleza**. 2024. Disponível em: <https://dados.fortaleza.ce.gov.br/dataset/groups/residuos-solidos-fortaleza>.

AKYON, F. C.; ALTINUC, S. O.; TEMIZEL, A. Slicing aided hyper inference and fine-tuning for small object detection. **2022 IEEE International Conference on Image Processing (ICIP)**, p. 966–970, 2022.

ANKILE, L. L.; HEGGLAND, M. F.; KRANGE, K. Deep convolutional neural networks: A survey of the foundations, selected improvements, and some current applications. **arXiv preprint arXiv:2011.12960**, 2020.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10004**: Resíduos sólidos: Classificação. Rio de Janeiro, 2004.

BEVANDIĆ, P.; KREŠO, I.; ORŠIĆ, M.; ŠEGVIĆ, S. Simultaneous semantic segmentation and outlier detection in presence of domain shift. In: SPRINGER. **Pattern Recognition: 41st DAGM German Conference, DAGM GCPR 2019, Dortmund, Germany, September 10–13, 2019, Proceedings 41**. [S.l.], 2019. p. 33–47.

BROWNLEE, J. **Deep learning for computer vision: image classification, object detection, and face recognition in python**. [S.l.]: Machine Learning Mastery, 2019.

CANTZLER, H. Random sample consensus (ransac). **Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh**, v. 3, 1981.

CARDOSO, F. d. C. A. I.; CARDOSO, J. C. O problema do lixo e algumas perspectivas para redução de impactos. **Ciência e Cultura**, scielocec, v. 68, p. 25 – 29, 12 2016. ISSN 0009-6725. Disponível em: <http://cienciaecultura.bvs.br/scielo.php?script=sci_arttext&pid=S0009-67252016000400010&nrm=iso>.

CóRDOVA, M.; PINTO, A.; HELLEVIK, C. C.; ALALIYAT, S. A.-A.; HAMEED, I. A.; PEDRINI, H.; TORRES, R. d. S. Litter detection with deep learning: A comparative study. **Sensors**, v. 22, n. 2, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/2/548>.

FOI, A.; TRIMECHE, M.; KATKOVNIK, V.; EGIAZARIAN, K. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. **IEEE transactions on image processing**, IEEE, v. 17, n. 10, p. 1737–1754, 2008.

FOUNDATION, L. D. I. **Wade-AI**. [S.l.]: GitHub, 2016. <https://github.com/letsdoitworld/wade-ai/tree/master/Trash_Detection>.

FRÄNTI, P.; MARIESCU-ISTODOR, R. Soft precision and recall. **Pattern Recognition Letters**, Elsevier, v. 167, p. 115–121, 2023.

FULTON, M. S.; HONG, J.; SATTAR, J. **Trash-ICRA19: A bounding box labeled dataset of underwater trash**. 2020.

GOMES, A. O. d. S.; BELéM, M. d. O. O lixo como um fator de risco À saÚde publica na cidade de fortaleza, cearÁ. **SANARE - Revista de Políticas Públicas**, v. 21, n. 1, jun. 2022. Disponível em: <https://sanare.emnuvens.com.br/sanare/article/view/1563>.

GONZALEZ, R. C. **Digital image processing**. [S.l.]: Pearson education india, 2009.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

HA, Q.; WATANABE, K.; KARASAWA, T.; USHIKU, Y.; HARADA, T. Mfnet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In: IEEE. **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2017. p. 5108–5115.

HAN, J.; SHAO, L.; XU, D.; SHOTTON, J. Enhanced computer vision with microsoft kinect sensor: A review. **IEEE transactions on cybernetics**, IEEE, v. 43, n. 5, p. 1318–1334, 2013.

HAO, S.; ZHOU, Y.; GUO, Y. A brief survey on semantic segmentation with deep learning. **Neurocomputing**, v. 406, p. 302–321, 2020. ISSN 0925-2312. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0925231220305476>.

HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. [S.l.]: Cambridge university press, 2003.

HE, K.; GKIOXARI, G.; DOLLáR, P.; GIRSHICK, R. **Mask R-CNN**. 2018. Disponível em: <https://arxiv.org/abs/1703.06870>.

HENDERSON, P.; FERRARI, V. End-to-end training of object class detectors for mean average precision. In: SPRINGER. **Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13**. [S.l.], 2017. p. 198–213.

HICKS, S. A.; STRÜMKE, I.; THAMBAWITA, V.; HAMMOU, M.; RIEGLER, M. A.; HALVORSEN, P.; PARASA, S. On evaluation metrics for medical applications of artificial intelligence. **Scientific reports**, Nature Publishing Group UK London, v. 12, n. 1, p. 5979, 2022.

HONG, J.; FULTON, M. S.; SATTAR, J. **TrashCan 1.0 An Instance-Segmentation Labeled Dataset of Trash Observations**. 2020.

JAIN, A. K. **Fundamentals of digital image processing**. [S.l.]: Prentice-Hall, Inc., 1989.

JOCHER, G. **YOLOv3 by Ultralytics**. 2020. Disponível em: <https://github.com/ultralytics/yolov3>.

JOCHER, G. **YOLOv5 by Ultralytics**. 2020. Disponível em: <https://github.com/ultralytics/yolov5>.

JOCHER, G.; CHAURASIA, A.; QIU, J. **Ultralytics YOLO V8**. 2023. Disponível em: <https://github.com/ultralytics/ultralytics>.

KIEFER, J.; WOLFOWITZ, J. Stochastic estimation of the maximum of a regression function. **The Annals of Mathematical Statistics**, JSTOR, p. 462–466, 1952.

KIRILLOV, A.; MINTUN, E.; RAVI, N.; MAO, H.; ROLLAND, C.; GUSTAFSON, L.; XIAO, T.; WHITEHEAD, S.; BERG, A. C.; LO, W.-Y.; DOLLáR, P.; GIRSHICK, R. **Segment Anything**. 2023. Disponível em: <https://arxiv.org/abs/2304.02643>.

KRAFT, M.; PIECHOCKI, M.; PTAK, B.; WALAS, K. Autonomous, onboard vision-based trash and litter detection in low altitude aerial images collected by an unmanned aerial vehicle. **Remote Sensing**, v. 13, n. 5, 2021. ISSN 2072-4292. Disponível em: <https://www.mdpi.com/2072-4292/13/5/965>.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LI, X.; WANG, W.; WU, L.; CHEN, S.; HU, X.; LI, J.; TANG, J.; YANG, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. **Advances in Neural Information Processing Systems**, v. 33, p. 21002–21012, 2020.

LIU, T.; STATHAKI, T. Faster r-cnn for robust pedestrian detection using semantic segmentation network. **Frontiers in neurorobotics**, Frontiers Media SA, v. 12, p. 64, 2018.

MAJCHROWSKA, S.; MIKOŁAJCZYK, A.; FERLIN, M.; KLAWIKOWSKA, Z.; PLANTYKOW, M. A.; KWASIGROCH, A.; MAJEK, K. Deep learning-based waste detection in natural and urban environments. **Waste Management**, Elsevier, v. 138, p. 274–284, 2022.

MANDHATI, S. R.; DESHAPRIYA, N. L.; MENDIS, C. L.; GUNASEKARA, K.; YRLE, F.; CHAKSAN, A.; SANJEEV, S. plitterstreet: Street level plastic litter detection and mapping. **arXiv preprint arXiv:2401.14719**, 2024.

Microsoft COCO Challenge Dataset. **Detection Evaluation**. 2022. https://cocodataset.org/detection-eval. Acessado em: 29/09/2024.

MUCELIN CARLOS ALBERTO BELLINI, M. Lixo e impactos ambientais perceptÍveis no ecossistema urbano. **Sociedade & Natureza**, 2008. ISSN 0103-1570. Disponível em: <https://www.redalyc.org/articulo.oa?id=321327192008>.

NETO, J. F. Aplicação da transformada de fourier no processamento digital de imagens. **Universidade de Aracaju-Se**, 1999.

PAGANI, F.; DELL'AMICO, M.; BALZAROTTI, D. Beyond precision and recall: understanding uses (and misuses) of similarity hashes in binary analysis. In: **Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy**. [S.l.: s.n.], 2018. p. 354–365.

PROENÇA, P. F.; SIMÕES, P. Taco: Trash annotations in context for litter detection. **arXiv preprint arXiv:2003.06975**, 2020.

RAVI, N.; GABEUR, V.; HU, Y.-T.; HU, R.; RYALI, C.; MA, T.; KHEDR, H.; RÄDLE, R.; ROLLAND, C.; GUSTAFSON, L.; MINTUN, E.; PAN, J.; ALWALA, K. V.; CARION, N.; WU, C.-Y.; GIRSHICK, R.; DOLLÁR, P.; FEICHTENHOFER, C. Sam 2: Segment anything in images and videos. **arXiv preprint arXiv:2408.00714**, 2024. Disponível em: <https://arxiv.org/abs/2408.00714>.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 779–788.

RODRIGUES, W.; FILHO, L. N. L. M.; PEREIRA, R. d. S. Análise dos determinantes dos custos de resíduos sólidos urbanos nas capitais estaduais brasileiras. **urbe. Revista Brasileira de Gestão Urbana**, Pontifícia Universidade Católica do Paraná, v. 8, n. 1, p. 130–141, Jan 2016. ISSN 2175-3369. Disponível em: <https://doi.org/10.1590/2175-3369.008.001.AO02>.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M. *et al.* Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, p. 211–252, 2015.

Samuel Pinusa. **Taxa de lixo de Fortaleza é aprovada na Câmara Municipal**. 2022. https://g1.globo.com/ce/ceara/noticia/2022/12/20/taxa-de-lixo-de-fortaleza-e-aprovada-na-camara-municipal.ghtml. Acessado em: 19/09/2024.

SORIA, X.; SAPPA, A.; HUMANANTE, P.; AKBARINIA, A. Dense extreme inception network for edge detection. **Pattern Recognition**, Elsevier, v. 139, p. 109461, 2023.

STOCKMAN, G.; SHAPIRO, L. G. **Computer vision**. [S.l.]: Prentice Hall PTR, 2001.

SZELISKI, R. **Computer vision: algorithms and applications**. [S.l.]: Springer Nature, 2022.

TERVEN, J.; CÓRDOVA-ESPARZA, D.-M.; ROMERO-GONZÁLEZ, J.-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. **Machine Learning and Knowledge Extraction**, MDPI, v. 5, n. 4, p. 1680–1716, 2023.

WANG, T.; CAI, Y.; LIANG, L.; YE, D. A multi-level approach to waste object segmentation. **Sensors**, v. 20, n. 14, 2020. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/20/14/3816>.

WU, Y.; KIRILLOV, A.; MASSA, F.; LO, W.-Y.; GIRSHICK, R. **Detectron2**. 2019. <https://github.com/facebookresearch/detectron2>.

XU, H.; YAN, Z.; JI, B.; HUANG, P.; CHENG, J.; WU, X. Defect detection in welding radiographic images based on semantic segmentation methods. **Measurement**, Elsevier, v. 188, p. 110569, 2022.

YANG, M.; THUNG, G. Classification of trash for recyclability status. **CS229 project report**, v. 2016, n. 1, p. 3, 2016.

ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. **Dive into Deep Learning**. [S.l.]: Cambridge University Press, 2023. <https://D2L.ai>.

ZHENG, Z.; WANG, P.; LIU, W.; LI, J.; YE, R.; REN, D. Distance-iou loss: Faster and better learning for bounding box regression. In: **Proceedings of the AAAI conference on artificial intelligence**. [S.l.: s.n.], 2020. v. 34, n. 07, p. 12993–13000.

ZHOU, D.; FANG, J.; SONG, X.; GUAN, C.; YIN, J.; DAI, Y.; YANG, R. Iou loss for 2d/3d object detection. In: IEEE. **2019 international conference on 3D vision (3DV)**. [S.l.], 2019. p. 85–94.