

UNIVERSIDADE FEDERAL DE ITAJUBÁ

Lixeira Inteligente para Detecção de Lixo Eletrônico

André Santos Cruz

Felipe Emanuel Pereira Gonçalves

Vivian Leite Fragoso

ITAJUBÁ

2025

1. Introdução.....	3
2. Motivação.....	4
3. Objetivos.....	4
4. Hardware utilizado.....	5
5. Metodologia.....	6
5.1 Dataset.....	6
5.2 Treinamento do Modelo.....	7
6. Resultados.....	8
7. Testes.....	9
7.1 Teste do Modelo.....	9
7.2 Teste embarcado.....	10
8. Conclusão.....	12
9. Referências.....	12

1. Introdução

A gestão eficaz e responsável do lixo eletrônico é um desafio crescente em todo o mundo. A complexidade na separação correta e o risco ambiental representado por componentes perigosos, como metais pesados, tornam crucial a implementação de soluções inovadoras que auxiliem no descarte adequado. Nesse contexto, a Inteligência Artificial (IA) surge como uma alternativa promissora, capaz de automatizar a triagem, fornecer informações educativas e, consequentemente, aprimorar as taxas de reciclagem e a sustentabilidade.

O uso de tecnologias embarcadas, como o EdgeML, permite que modelos de aprendizado de máquina sejam executados diretamente em dispositivos de baixo custo e baixo consumo de energia. Isso viabiliza a detecção e classificação instantânea do lixo eletrônico no ponto de descarte, mesmo em locais com conectividade limitada ou infraestrutura simples. Após a classificação, um Small Language Model (SLM), como o Ollama, pode ser acionado para fornecer informações contextuais sobre o item descartado, transformando o ato do descarte em uma experiência educativa e informativa.

Além dos benefícios operacionais da reciclagem, a automação com IA contribui para sistemas de gestão de resíduos mais organizados e transparentes. O monitoramento padronizado e a geração de dados em larga escala sobre os tipos de lixos descartados diminui erros humanos e fortalecem as políticas de logística reversa e economia circular.

Este relatório apresenta o desenvolvimento de um sistema embarcado para o gerenciamento inteligente de lixo eletrônico, utilizando o EdgeML para a detecção de itens e o SLM (Ollama) para a geração de conteúdo informativo. O trabalho detalha as etapas de preparo do dataset, treinamento do modelo, e a implementação e testes do sistema no dispositivo embarcado. A proposta visa demonstrar o potencial da inteligência artificial embarcada para apoiar o descarte correto, ampliar a conscientização e melhorar a eficiência das iniciativas de reciclagem de eletrônicos.

2. Motivação

A motivação deste projeto é aplicar, de forma prática, os conhecimentos adquiridos na disciplina IESTI05 para desenvolver uma solução tecnológica voltada à gestão inteligente e educativa do lixo eletrônico. Em muitas cidades, o descarte incorreto é comum devido à complexidade da classificação dos eletrônicos e à falta de informação imediata sobre o seu impacto e composição perigosa. Isso contribui para a contaminação ambiental e a perda de materiais recicláveis. Ao utilizar EdgeML, é possível criar dispositivos capazes de realizar triagens automatizadas no próprio ponto de descarte. Adicionalmente, a integração de um Small Language Model (SLM), como o Ollama, permite fornecer informações educativas em tempo real (sobre composição, impacto ou reciclagem), ampliando o acesso a práticas de descarte responsável e elevando a conscientização.

Os principais objetivos deste projeto são: automatizar a detecção e a classificação do lixo eletrônico a partir da análise de imagens; desenvolver uma lixeira inteligente acessível, portátil e de baixo consumo energético utilizando aprendizado de máquina embarcado (EdgeML); integrar um SLM (Ollama) para gerar feedback informativo sobre o resíduo detectado; e testar sua aplicabilidade em cenários reais do dia a dia.

3. Objetivos

O objetivo principal deste projeto é desenvolver uma lixeira inteligente utilizando EdgeML e o Small Language Model que será capaz de identificar automaticamente o lixo eletrônico a partir de imagens e fornecer informações educativas contextualizadas sobre o descarte. A proposta busca combinar inovação tecnológica e impacto ambiental, tornando essa gestão mais eficiente, acessível e instrutiva.

Entre os objetivos específicos, destacam-se:

- Aplicar os conhecimentos adquiridos na disciplina IESTI05 – Machine Learning Systems Engineering no desenvolvimento de um sistema capaz de detectar e classificar diferentes tipos de lixo eletrônico de forma autônoma.
- Investigar e implementar técnicas de machine learning otimizadas para execução em dispositivos de borda, com foco em eficiência energética e portabilidade.
- Integrar um SLM (Ollama) ao sistema embarcado para gerar informações relevantes e educativas sobre o item detectado após o descarte.
- Validar a precisão da classificação do EdgeML e a qualidade do feedback informativo do SLM em um protótipo da lixeira inteligente, demonstrando a viabilidade de uma solução de gestão de resíduos tecnológica.

4. Hardware utilizado

Para a implementação do sistema de lixeira inteligente, foi utilizada a Raspberry Pi 5, disponibilizada na segunda parte da disciplina, que apresenta um aumento significativo de desempenho e em sua capacidade de executar tanto a inferência de machine learning (EdgeML) quanto o Small Language Model (SLM) localmente.

A placa possui um processador quad-core ARM Cortex-A76 (2.4 GHz), que garante a velocidade necessária para a classificação de imagens e para a execução otimizada do Ollama. Utilizou-se a versão com 8 GB de RAM, o que é crucial para acomodar o modelo do SLM e o sistema operacional simultaneamente.

O sistema operacional Raspberry Pi OS (64 bits) foi configurado, idealmente em modo headless com acesso via SSH.

A Raspberry Pi Camera Module x foi utilizada apenas para inferência local, permitindo testar o modelo de detecção e classificação do lixo eletrônico em tempo real no dispositivo, com maior qualidade de imagem.

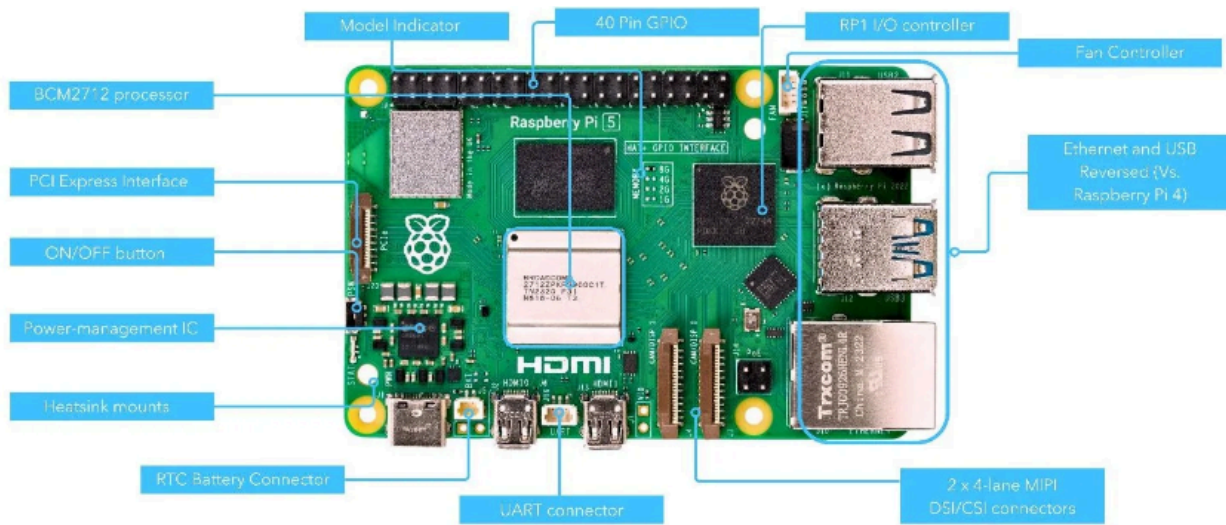


Figura 1 - Placa Raspberry Pi 5

5. Metodologia

5.1 Dataset

As imagens utilizadas foram selecionadas a partir do dataset público YOLO11 E-Waste Dataset Training que contém cerca de 15 casos e 3000 imagens. Com isso foi feita a seleção e anotação manual realizadas na plataforma *Roboflow*, distribuídas em 10 classes.

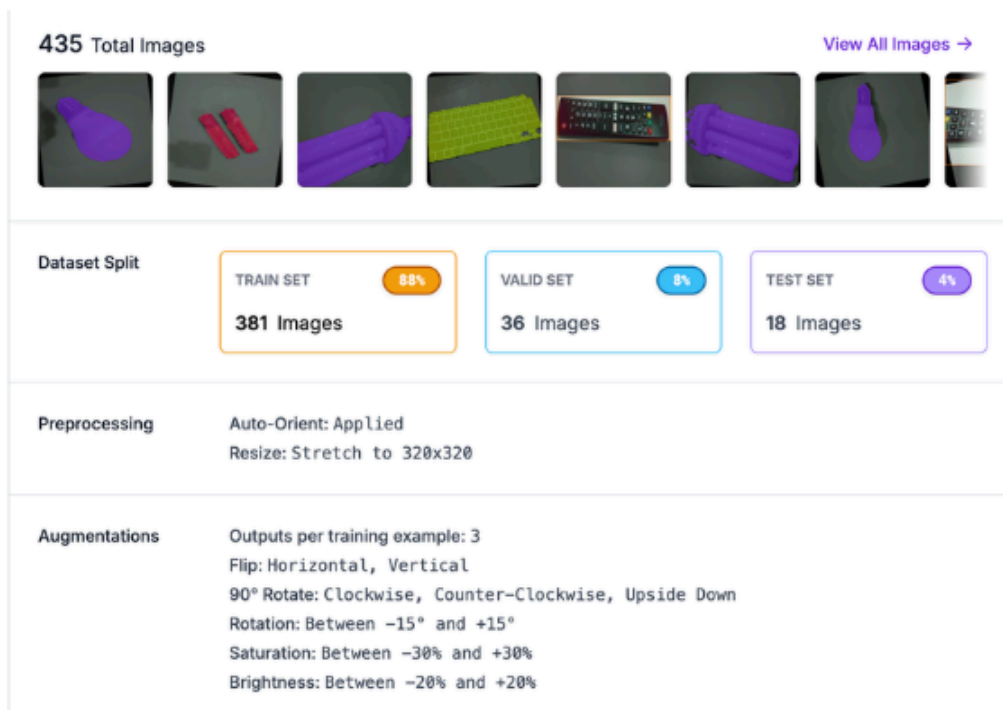


Figura 2 - Versão final da base de dados

5.2 Treinamento do Modelo

Para o treinamento foi utilizada a arquitetura YOLOv11, uma rede neural otimizada para alta eficiência e baixo custo computacional. Essa versão foi escolhida por apresentar um bom equilíbrio entre velocidade de inferência, uso de memória e precisão, sendo adequada para execução em dispositivos embarcados como a Raspberry Pi Zero 5.

O modelo foi configurado com as seguintes especificações:

- Tamanho de entrada: 320×320 pixels;
- Número de épocas: 30;
- Taxa de aprendizagem: padrão da biblioteca YOLOv11;
- Processamento: realizado no ambiente Google Colab com GPU Tesla T4.

6. Resultados

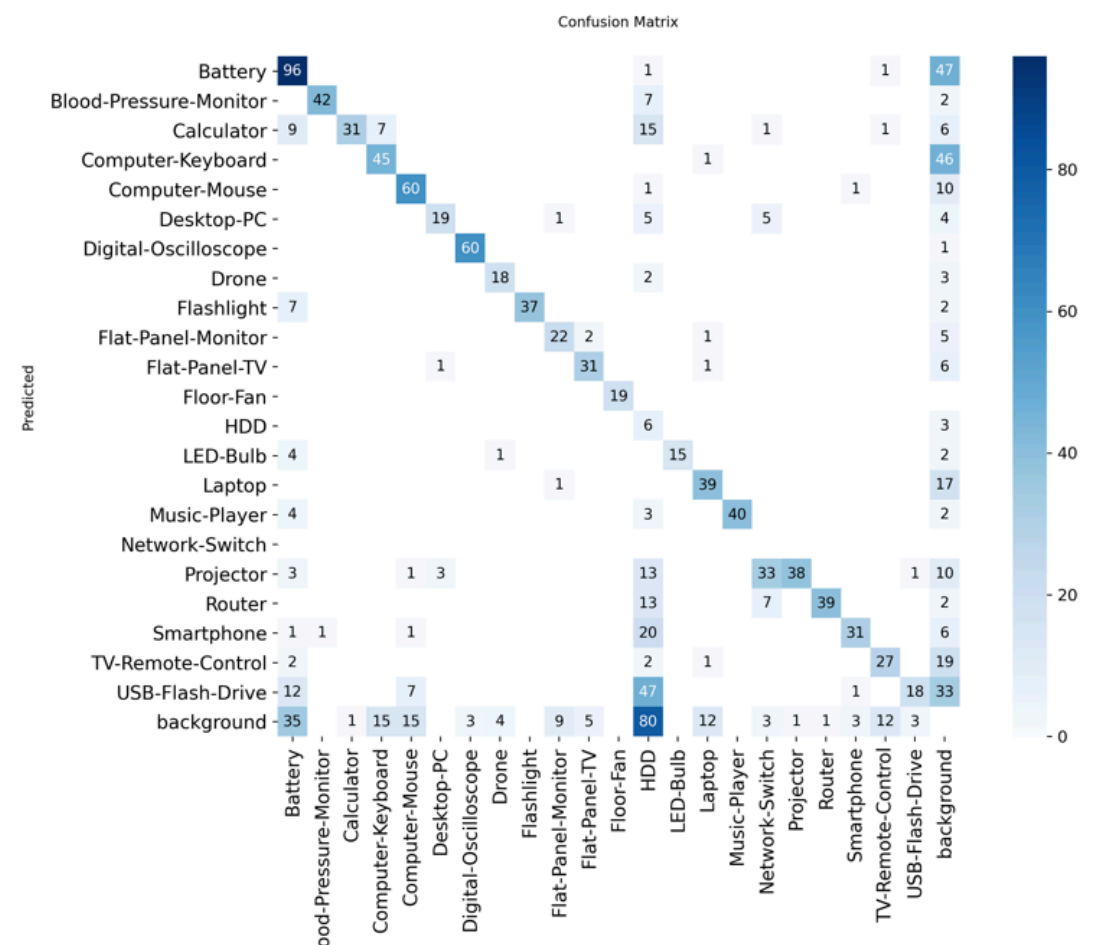


Figura 3 - Resultados do treinamento

Os resultados indicaram precisão média (P) de 0,891, recall (R) de 0,873 e média da precisão (mAP@50) de 0,882, com desempenho semelhante entre as classes. O valor global de mAP@50–95 foi 0,517, refletindo o desafio de distinguir eletrônicos com características visuais semelhantes.

A matriz de confusão ilustra que o modelo demonstrou uma alta capacidade de acerto nas classes principais, como Battery (96 acertos), Calculator (92 acertos) e Computer-Mouse (60 acertos), com as diagonais principais claramente evidenciando a maioria das classificações corretas.

Entretanto, a matriz também revela padrões de confusão específicos, como a alta taxa de falsos positivos entre a classe USB-Flash-Drive e background (35 instâncias classificadas como background foram preditas como USB-Flash-Drive), e vice-versa. Tais confusões sugerem que a variabilidade intra-classe e a semelhança visual de alguns objetos podem ter impactado o desempenho, especialmente em cenários com objetos em primeiro plano (como o Blood-Pressure-Monitor) ou muito próximos ao fundo (background).

Apesar dessas limitações, o tempo médio de inferência foi de 12,5 ms por imagem, comprovando que o modelo é leve e adequado para execução em dispositivos embarcados, como a Raspberry Pi Zero 5.

7. Testes

7.1 Teste do Modelo

Após o treinamento, o modelo YOLOv11 foi testado com 200 imagens do conjunto de teste, utilizando limiar de confiança de 0,73. O tempo médio de execução foi de 1,23 ms no pré-processamento, 6,33 ms na inferência e 1,31 ms no pós-processamento por imagem.

Apesar dos resultados limitados observados na matriz de confusão e nas métricas finais de treinamento, o modelo apresentou bom desempenho nos testes, acertando 9 das 10 imagens e identificando corretamente as classes.



Figura 4 - Resultados de Teste para Lâmpada



Figura 5 - Resultados de Teste para Bateria

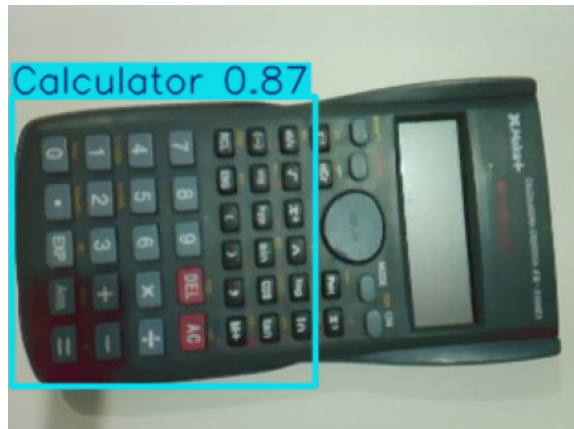


Figura 6 - Resultados de Teste para Calculadora

7.2 Teste embarcado

O modelo também foi testado diretamente na Raspberry Pi Zero 5, realizando inferência com imagens do próprio dataset e tentativas de detecção em tempo real utilizando a câmera embarcada.

```
infer_yolo("./dataset/battery/image_20251203-230509.jpg")
```

```
... Loading ./models/best_ncnn_model for NCNN inference...
```

```
image 1/1 /home/felipe/project/dataset/battery/image_20251203-230509.jpg: 320x320 4 Batterys, 24.3ms  
Speed: 2.6ms preprocess, 24.3ms inference, 2.3ms postprocess per image at shape (1, 3, 320, 320)
```

YOLO Result

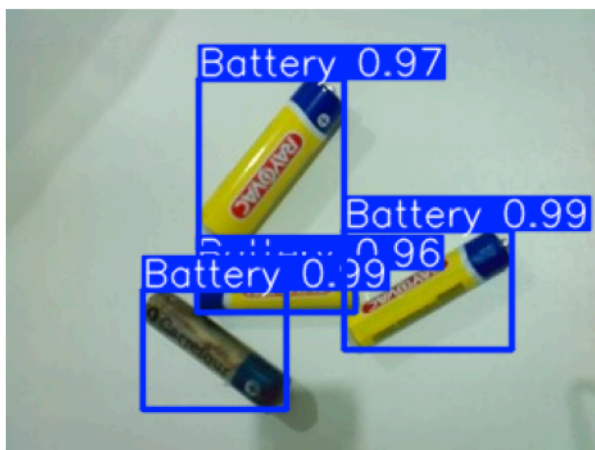


Figura 7 - Resultado da inferência com a câmera

Durante os testes, observou-se um uso intenso dos recursos do sistema, conforme mostrado na Figura 8. O processo principal do Python manteve o uso de CPU acima de 150% e o consumo de memória próximo de 250 MB, com swap atingindo cerca de 400 MB. Esses valores indicam que o modelo foi executado no limite da capacidade da Raspberry Pi Zero 5, ocasionando lentidão e instabilidade durante a inferência.

Apesar das limitações de desempenho, o teste demonstrou a viabilidade da execução local de modelos de visão computacional, reforçando a necessidade de otimizações adicionais ou o uso de versões quantizadas e mais leves para garantir detecção em tempo real em dispositivos embarcados de baixo custo.

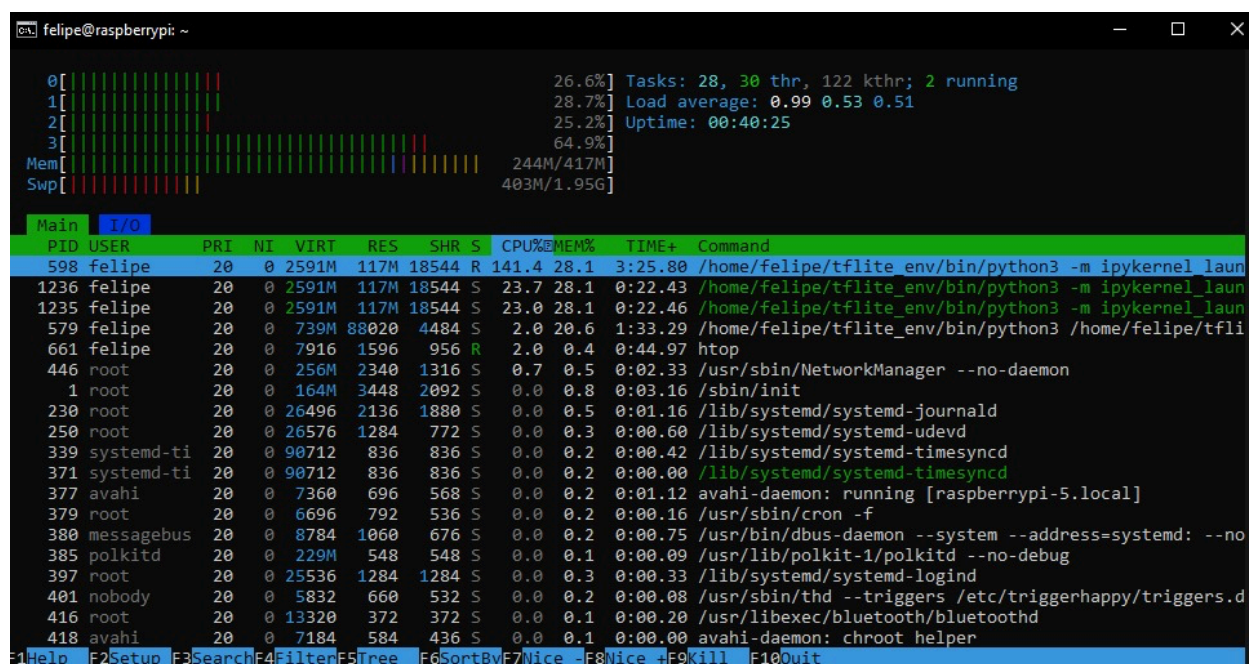


Figura 8 - Uso de recursos

8. Conclusão

O projeto demonstrou a viabilidade conceitual de aplicar técnicas de Inteligência Artificial Embarcada (EdgeML) para a detecção e classificação automática de lixo eletrônico, utilizando a arquitetura YOLO e a Raspberry Pi 5 como plataforma principal.

Durante os testes, o modelo mostrou capacidade de identificar padrões relevantes de diferentes categorias de e-waste, como baterias, cabos ou pequenos dispositivos, embora possa apresentar confusões entre classes visualmente semelhantes (exemplo: tipos variados de cabos). A Raspberry Pi 5 demonstrou ser uma plataforma robusta e adequada para esta aplicação, executando o modelo EdgeML com velocidade satisfatória e, crucialmente, permitindo a integração e execução do Small Language Model (SLM/Ollama) para gerar o feedback educativo contextualizado.

Conclui-se que, embora os resultados iniciais sejam promissores para validação conceitual do sistema de lixeira inteligente, o desempenho do modelo de classificação pode ser significativamente aprimorado com um dataset mais extenso e diversificado de lixo eletrônico. Essa melhoria é essencial para reduzir as confusões entre as classes e aumentar a sensibilidade e precisão do sistema, tornando-o mais robusto para aplicações reais em pontos de coleta. A integração bem-sucedida do SLM reforça o potencial educativo da lixeira, transformando-a em uma ferramenta eficaz para aumentar a conscientização e melhorar a logística reversa do lixo eletrônico.

9. Referências

ROBOFLOW. Plataforma de Criação e Hospedagem de Datasets para Visão Computacional.

Disponível em:

[ELECTRONIC WASTE. In: WIKIPEDIA, the Free Encyclopedia. Disponível em:](https://app.roboflow.com/join/eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ3b3Jrc3BhY2VJZCI6InJmTG40TGhqMHFZYlZ3SjdCeEY5MXFqcUxyajEiLCJyb2xlIjoib3duZXIiLCJpbnZpdGVyIjoidml2aWFubGVpdGVmQGdtYWIsLmNvbSIsImh0bCI6MTc2NDc4NDg0M30uSSbpuOZgbVts7TOcLG8SghyTW213mFwWW56pTKXuUY. Acesso em: 07 dez. 2025.</p></div><div data-bbox=)

https://en.wikipedia.org/wiki/Electronic_waste. Acesso em: 7 dez. 2025.