

# CLASSIFICAÇÃO DOS RESÍDUOS RECICLÁVEIS

## ETAPA 1 – Análise e Preparação do dataset WARP

### Squad:

Ana Sofia  
Elineison Inacio  
Felipe Miguel  
Franciscleide Lauriano  
Iza Francine  
Madelu Lopes  
Mariana Angeli  
Rodrigo Luiz

# Objetivos



Avaliar a integridade, consistência e qualidade dos dados antes da modelagem

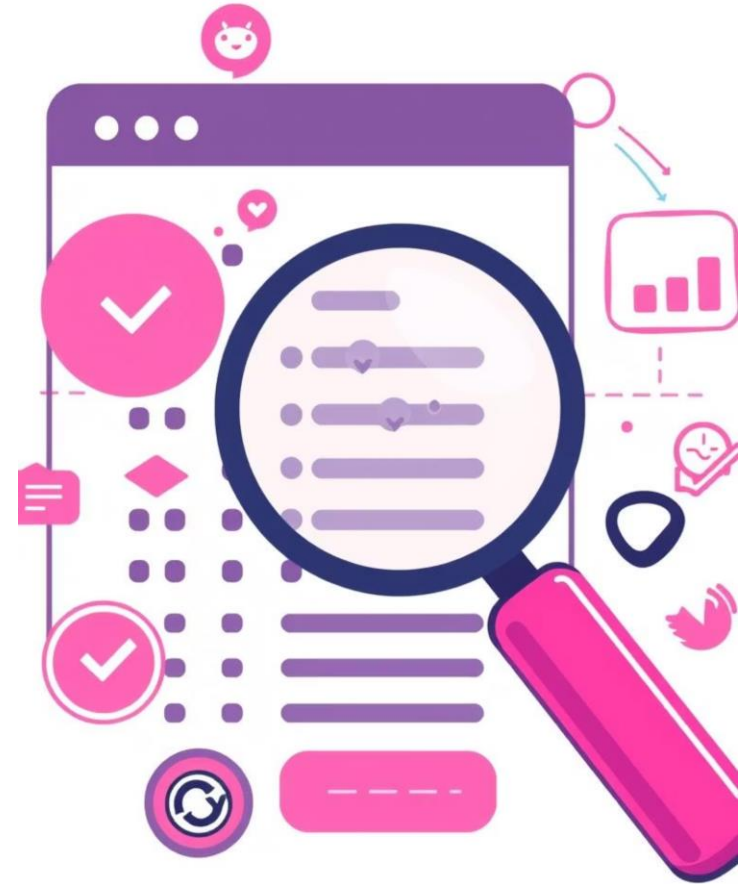


Verificar e tratar problemas como:

- Arquivos corrompidos
- Valores ausentes ou inconsistentes
- Outliers nas dimensões
- Imagens duplicadas
- Desequilíbrio entre classes



Preparar o dataset para uso em modelos de machine learning



# Metodologia



## Análise Exploratória

Foi realizada uma **análise exploratória do dataset** com o objetivo de avaliar a integridade, consistência e qualidade dos dados.



## Fonte

A **fonte do dataset** utilizado foi o **Warp Waste Recycling Plant Dataset**, disponibilizado na plataforma **Kaggle** pelo usuário parohod:

🔗 <https://www.kaggle.com/datasets/parohod/warp-waste-recycling-plant-dataset>



# Metodologia

---



## Integridade dos Arquivos

1. **Detectar Arquivos órfãos ou registros inválidos:** Verificação da correspondência entre arquivos físicos e caminhos listados no CSV ('image\_path' no DataFrame df).
2. **Verificação do formato das imagens:** Avaliação do formato das imagens (.jpg, .png etc) usando a coluna 'image\_format'.



## Consistência dos Metadados

1. **Inspeção de valores nulos** com `isnull().sum()` da bib. Pandas.
2. Estatísticas com `describe()` para `width`, `height` e `channels`.
3. **Validação da uniformidade dos canais** (todos 3 = RGB) e **Deteção de outliers** com IQR.



## Qualidade das Imagens

1. **Verificação de imagens corrompidas:** Leitura de todas as imagens com `PIL.Image.open()` da bib. PIL.
2. Tratamento de exceções para identificar arquivos corrompidos ou inacessíveis

# Metodologia

---



## Distribuição das Classes

1. Investigar distribuição das classes: Análise com `value_counts()` da bib. pandas e visualização com gráfico de barras para identificar desequilíbrio.



## Verificação de Duplicatas

Identificar duplicações:

1. Geração de **hash perceptual** com `'imagehash.phash()'` e **hash MD5** a partir do conteúdo binário real do arquivo usando a bib. `hashlib`.
2. Detecção de imagens idênticas com `.duplicated(keep=False)`
3. Verificação de duplicidade nos registos do CSV

# Sobre o Dataset

## Nome - Fonte

Warp Waste Recycling Plant Dataset

Kaggle: [parohod/warp-waste-recycling-plant-dataset](https://www.kaggle.com/parohod/warp-waste-recycling-plant-dataset)

## Conteúdo

Imagens de resíduos recicláveis categorizadas por tipo.

## Categorias

Papel, vidro, metal, plástico, entre outros.

## Tamanho Inicial

8.823 imagens em 28 classes.



# Resultados

---



## Integridade

- Correspondência total entre CSV e diretório
- Nenhuma imagem ausente
- 100% das imagens em formato .jpg



## Qualidade

Nenhuma imagem corrompida ou inacessível foi encontrada



## Duplicatas

- Nenhuma imagem duplicada visualmente
- Nenhuma duplicidade de nome no arquivo CSV

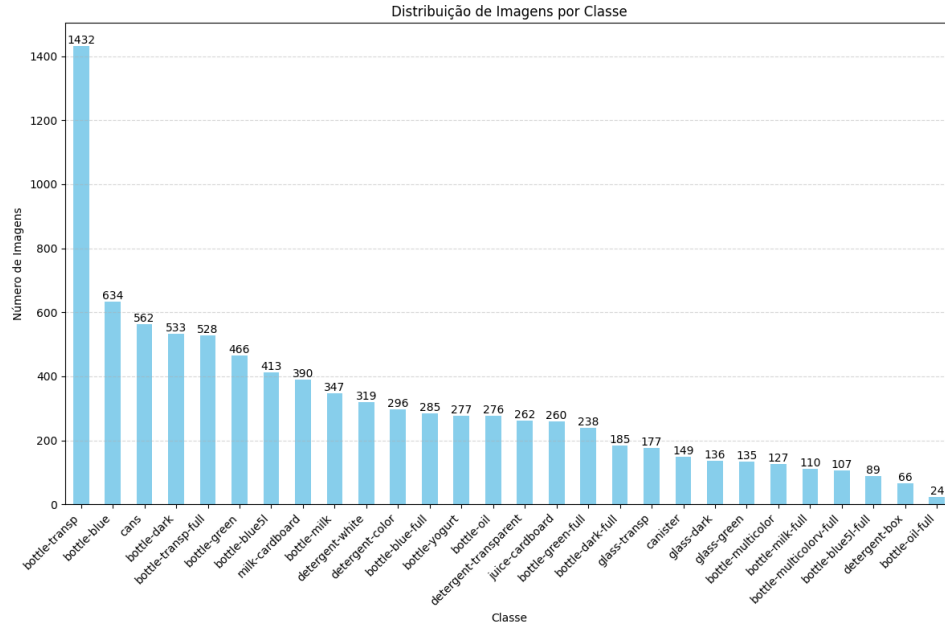
# Resultados



## Distribuição das Classes

### Desequilíbrio severo identificado:

- O Classe mais frequente: bottle-transp (1.432 imagens)
- O Classe menos frequente: bottle-oil full (24 imagens)
- O A maioria das classes com menos de 300 amostras



**Risco de viés nos modelos:** comprometer o desempenho, especialmente o recall das classes minoritárias




# Resultados



## Consistência dos Metadados

- Nenhum valor ausente nos metadados
- Todos os valores de channels iguais a 3 (RGB)
- Foram identificados outliers significativos:
  - 212 imagens com width > 400
  - 416 com height > 383.5

 Estatísticas descritivas iniciais das colunas de dimensão:

	width	height	channels
count	8823.000000	8823.000000	8823.0
mean	174.195172	182.712909	3.0
std	90.048484	92.396158	0.0
min	35.000000	40.000000	3.0
25%	105.000000	116.000000	3.0
50%	150.000000	159.000000	3.0
75%	223.000000	223.000000	3.0
max	668.000000	703.000000	3.0

Cálculo dos limites para outliers com IQR (antes de qualquer tratamento, se necessário):

Coluna: WIDTH  
IQR: 118.0  
Limite Inferior: -72.0  
Limite Superior: 400.0  
Total de outliers em width: 212

Coluna: HEIGHT  
IQR: 107.0  
Limite Inferior: -44.5  
Limite Superior: 383.5  
Total de outliers em height: 416

Coluna: CHANNELS  
IQR: 0.0  
Limite Inferior: 3.0  
Limite Superior: 3.0  
Total de outliers em channels: 0

# Resultados

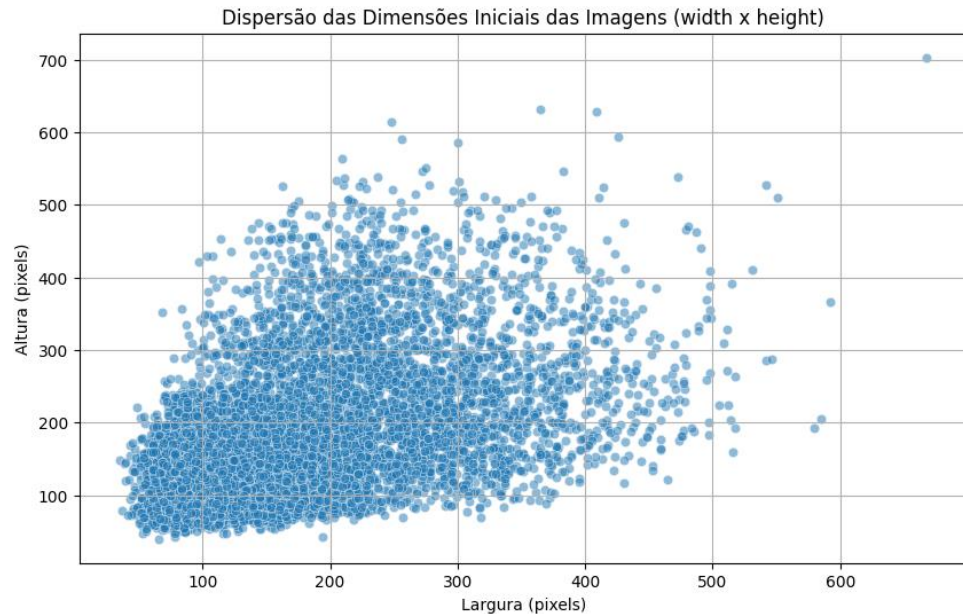


## Consistência dos Metadados



### Riscos causados por outliers:

- Dificultam a padronização das imagens;
- Podem gerar distorções visuais ao redimensionar;
- Prejudicam o desempenho de modelos sensíveis à escala;
- Aumentam o risco de overfitting.



# Recomendações para etapa de Pré-Processamento

---

## Desequilíbrio entre Classes

- Aplicar Data Augmentation nas classes minoritárias (ex: rotação, brilho, zoom).
- Utilizar técnicas de Oversampling como SMOTE ou ADASYN nos embeddings de CNNs.
- Combinações híbridas (ex: SMOTE + Tomek Links) podem melhorar generalização.
- Monitorar F1-score e Recall por classe para validar o impacto do balanceamento.
- Repetir a análise sempre que o dataset for atualizado.

## Outliers nas Dimensões

- Redimensionar para 224x224 com Resize + Padding para padronizar entradas.
- Alternativas: Resize com Crop Central ou técnicas de Letterbox.
- Revalidar metadados pós-redimensionamento.
- Armazenar imagens redimensionadas em nova pasta organizada por classe.
- Incorporar essa análise ao pipeline contínuo de ingestão de dados.

# Conclusões



## Pontos Positivos

- ▶ Dados bem organizados e consistentes
- ▶ Ausência de imagens corrompidas ou duplicadas
- ▶ Classes bem definidas semanticamente



## Desafios

- ▶ Desequilíbrio acentuado entre classes
- ▶ Variação extrema nas resoluções das imagens



## Ações Recomendadas

- ▶ Balanceamento com Augmentation/Oversampling
- ▶ Padronização com resize + padding

# Próximos Passos

---



## Aplicar Estratégias para Balancear Classes

Aplicar técnicas de augmentations específicas nas classes minoritárias.  
Avaliar impacto de SMOTE ou ADASYN nos embeddings para balanceamento.



## Padronizar Dimensões de Imagem

Redimensionar imagens para 224x224 com resize + padding preservando proporção.



## Reaplicar análises

Reaplicar análises exploratórias e validações caso novas imagens sejam incorporadas ao dataset.

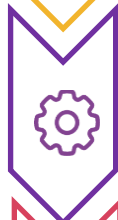
# Próximos Passos

---



## Selecionar Arquiteturas Base

Iniciar testes com EfficientNet-B0 e ResNet-50.

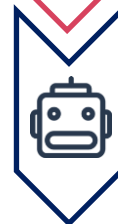


## Ajustar Transferência de Aprendizado

Congelar camadas base e ajustar top layers no fine-tuning.



## Monitorar métricas por classe durante o treinamento



## Automatizar pipeline

Automatizar o pipeline de tratamento de dados, para garantir reprodutibilidade e rastreabilidade nas etapas futuras do projeto.

# Referências Bibliográficas

---

- Dosovitskiy, A. et al. (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv. Disponível em: <https://arxiv.org/abs/2010.11929>
- He, K.; Zhang, X.; Ren, S.; Sun, J. (2016). *Deep Residual Learning for Image Recognition*. CVPR. Disponível em: <https://arxiv.org/abs/1512.03385>
- Huang, G. et al. (2017). *Densely Connected Convolutional Networks*. CVPR. Disponível em: <https://arxiv.org/abs/1608.06993>
- PAROHOD. *WaRP – Waste Recycling Plant Dataset*. Kaggle, 2022. Disponível em: <https://www.kaggle.com/datasets/parohod/warp-waste-recycling-plant-dataset>
- Sandler, M. et al. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. CVPR. Disponível em: <https://arxiv.org/abs/1801.04381>
- Simonyan, K.; Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv. Disponível em: <https://arxiv.org/abs/1409.1556>
- Szegedy, C. et al. (2016). *Rethinking the Inception Architecture for Computer Vision*. CVPR. Disponível em: <https://arxiv.org/abs/1512.00567>
- Tan, M.; Le, Q. V. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. ICML. Disponível em: <https://arxiv.org/abs/1905.11946>
- Tan, M.; Le, Q. V. (2021). *EfficientNetV2: Smaller models and faster training*. ICML. Disponível em: <https://arxiv.org/abs/2104.00298>
- WaRP Dataset. MIT License, 2025. Disponível em: <https://opensource.org/licenses/MIT>