UNIVERSIDADE FEDERAL DE ALAGOAS

Novo Ensino Suplementar

Classificação de Lixo com Redes Neurais: Uma Análise Comparativa entre MLP e CNN

Aluno: Almir Sérgio Ramos dos Santos Filho
Disciplina: Redes Neurais - NES
Professor: Eduardo Adame
Link do repositório: https://github.com/oalmirsergio/midterm-nn-avaliacaol

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

1. Introdução

Este trabalho tem como objetivo desenvolver e validar modelos de aprendizado profundo para classificação automática de 12 categorias de lixo.

O projeto foi estruturado sob a restrição de no máximo 1 milhão de parâmetros, garantindo eficiência e robustez.

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

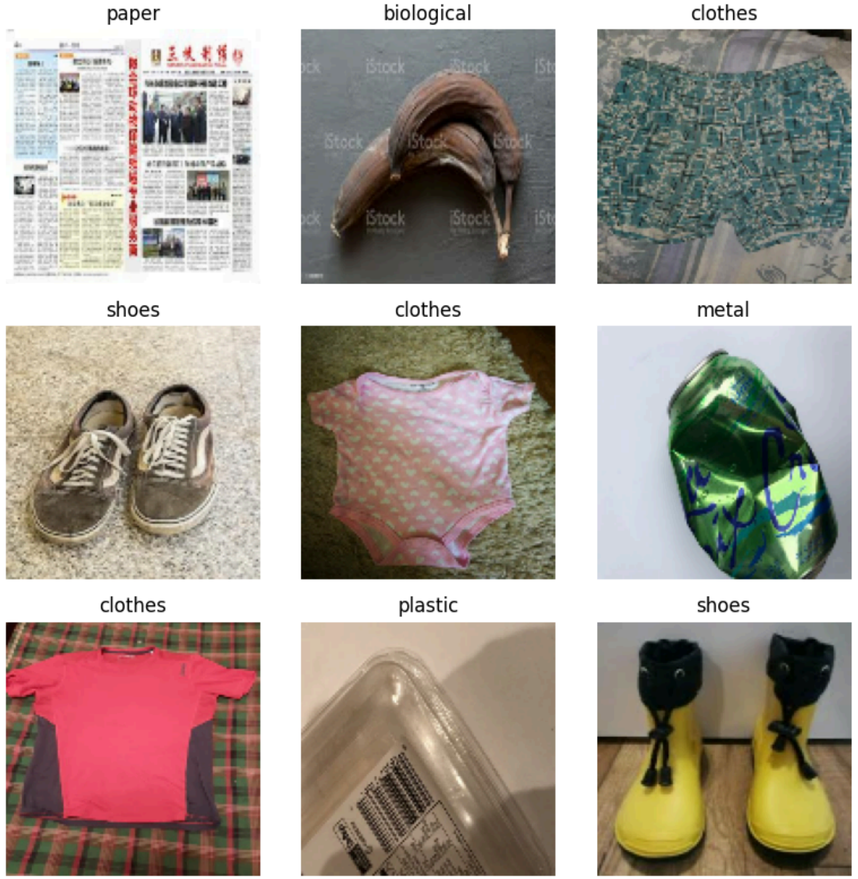
Problema: A necessidade de automação na triagem de lixo para reciclagem. Como podemos classificar imagens de diferentes tipos de resíduos de forma eficiente?

Objetivo: Investigar e comparar a eficácia de duas arquiteturas de redes neurais (MLP e CNN) na tarefa de classificação de imagens de lixo.

Hipótese Central: A arquitetura de Redes Neurais Convolucionais (CNN), por sua capacidade de extrair features espaciais, apresentará um desempenho significativamente superior ao de uma rede Perceptron de Múltiplas Camadas (MLP) na classificação das imagens do dataset.

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar



UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

2. Objetivos e Restrições

- Desenvolver uma CNN eficiente para classificar corretamente diferentes tipos de lixo.
- Comparar duas arquiteturas: MLP (baseline) e CNN (otimizada).
- Respeitar a restrição de 1 milhão de parâmetros, com foco em robustez estatística.
- Utilizar Grid Search 2×2×3 (12 experimentos) para garantir confiabilidade.

UNIVERSIDADE FEDERAL DE ALAGOAS

Novo Ensino Suplementar

3. Metodologia

O projeto foi dividido em blocos de implementação em Python, cada um com um propósito específico.

Bloco 1: Definição de Parâmetros Globais

- EPOCHS = 30: garante comparação uniforme.
- SEEDS = [42, 43, 44]: validação estatística.
- BATCH_SIZE = 64 e IMG_SIZE = (128,128). e IMG_SIZE_MLP = (48,48)

Bloco 2: Pré-processamento e Feature Engineering

- Funções de normalização (scale_images).
- resize_only: baseline.
- central_crop_and_resize: corte central (70%) para remover ruído do fundo.

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

Bloco 3: Organização dos Dados

- Divisão estratificada 80/20 (treino/teste).
- Seed fixa para garantir reprodutibilidade.
- Uso de pastas train/ e test/ para cada classe.

Bloco 4: Otimização do Pipeline de Dados

- Uso de .map().cache().prefetch(AUTOTUNE).
- Elimina gargalos de I/O e garante eficiência do treinamento.

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

4. Arquiteturas e Treinamento

- MLP (Baseline): input 48x48, menos parâmetros mas menos informação.
- CNN (Otimizada): input 128x128, uso de GlobalAveragePooling2D para reduzir parâmetros.
- Regularização: Dropout e L2 para evitar overfitting.
- Restrição verificada: assert params <= 1_000_000.

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

5. Avaliação

Função train_and_evaluate:

- Define treino e avaliação padronizados.
- Métrica principal: F1-Macro (considera todas as classes igualmente).
- Gera matriz de confusão e relatórios de classificação.

UNIVERSIDADE FEDERAL DE ALAGOAS

Novo Ensino Suplementar

Tabela de Resultados

A	B	C	D	E
Modelo	Pré-processame	Acurácia (Média	F1-Macro (Média ± DP)	
CNN	Resize	0.735 ± 0.002	0.675 ± 0.004	
CNN	Crop	0.476 ± 0.020	0.271 ± 0.029	
MLP	Resize	0.395 ± 0.046	0.135 ± 0.096	
MLP	Crop	0.418 ± 0.031	0.169 ± 0.066	

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

7. Limitações e Trade-offs

- Não foi usado Early Stopping (treino fixo em 30 épocas).
- Não foi aplicado Transfer Learning (restrição de <1M parâmetros).
- Dataset desbalanceado prejudicou classes minoritárias (ex: Metal, Plastic).

UNIVERSIDADE FEDERAL
DE ALAGOAS

Novo
Ensino
Suplementar

8. Próximos Passos

- Implementar Weighted Loss ou Focal Loss para equilibrar classes minoritárias.
- Aplicar Data Augmentation específico em classes com menos exemplos.
- Usar Early Stopping em produção para evitar overfitting.
- Explorar arquiteturas híbridas (CNN + Vision Transformer leve).

OBRIGADO