

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

**Autor:** Almir Sérgio Ramos dos Santos Filho

**Professor Orientador:** Prof. Eduardo Adame

**Curso:** Redes Neurais - NES

**Tema:** Comparação entre CNN e MLP na classificação de resíduos sólidos (dataset TrashNet)

## 1 Pergunta / Hipótese

Hipótese: Redes Convolucionais (CNNs), mesmo sob restrição de até 1 milhão de parâmetros, apresentam desempenho superior a perceptrons multicamadas (MLPs) na tarefa de classificação de 12 categorias de lixo.

## 2 Dados

O dataset utilizado é o **TrashNet**, amplamente usado em tarefas de classificação de resíduos. Ele contém aproximadamente **10.000 imagens**, distribuídas em 12 classes: *battery, biological, brown-glass, cardboard, clothes, green-glass, metal, paper, plastic, shoes, trash, white-glass*.

A divisão adotada foi:

- Treino: 80% (com estratificação por classe);
- Teste: 20% (sem embaralhamento).

Pré-processamento realizado:

- Normalização dos pixels para  $[0, 1]$ ;
- Redimensionamento para  $128 \times 128$  (CNN) e  $48 \times 48$  (MLP);
- Testes com *resize only* e *central crop*.

## 3 Modelos

**MLP:** entrada  $48 \times 48$ , camada **Flatten**, seguida de densas totalmente conectadas (**Dense**), ativação **ReLU**, regularização **Dropout**, e saída **Softmax** com 12 neurônios.

**CNN:** entrada  $128 \times 128$ , múltiplas camadas **Conv2D** com filtros  $3 \times 3$  e ativação **ReLU**, **MaxPooling**, **GlobalAveragePooling2D** para reduzir parâmetros, regularização **Dropout** e penalização **L2**, e saída **Softmax**.

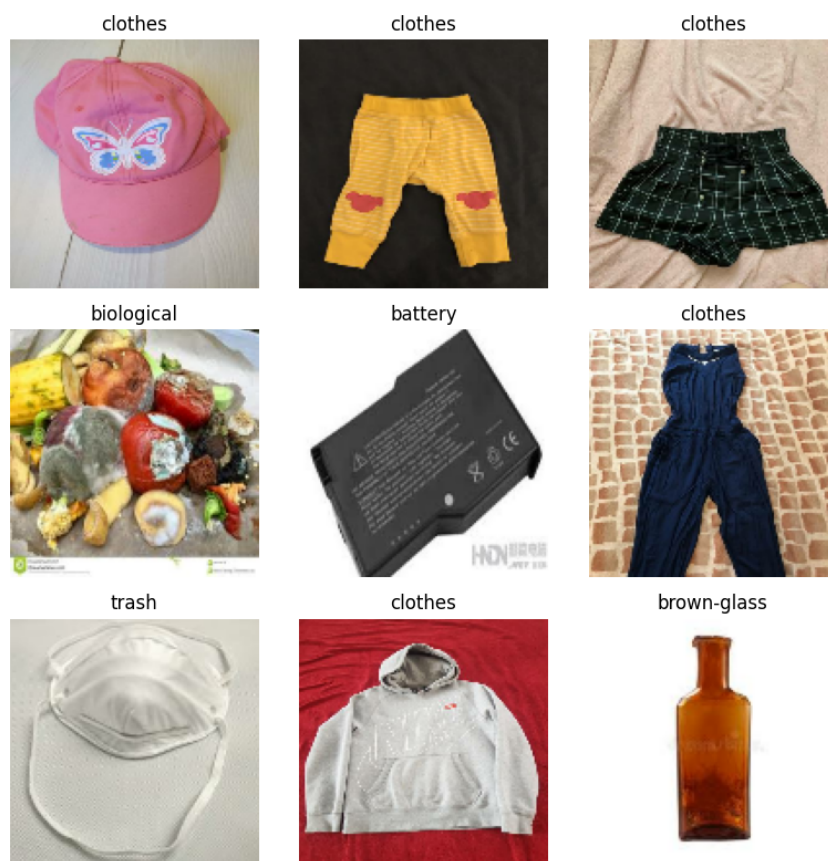


Figura 1: Exemplos de imagens por classe.

## 4 Treinamento

- Otimizador: Adam, taxa de aprendizado padrão ( $10^{-3}$ );
- Épocas: 30 (fixo, sem early stopping);
- Função de perda: Categorical Cross-Entropy;
- Regularização: Dropout (0.5) e L2;
- Seeds: 42, 43 e 44 (reprodutibilidade).

## 5 Experimentos

O desenho experimental utilizou uma grade (*grid search*) de combinações:

- Modelos: CNN e MLP;
- Pré-processamento: Resize e Crop;
- Seeds: 42, 43, 44.

Total de 12 experimentos.

As métricas usadas foram: **Acurácia** e **F1-macro**.

Tabela 1: Resultados obtidos nos experimentos (média  $\pm$  desvio padrão).

Modelo	Acurácia	F1-macro
CNN (Resize)	$0.731 \pm 0.007$	$0.666 \pm 0.009$
CNN (Crop)	$0.463 \pm 0.010$	$0.253 \pm 0.021$
MLP (Resize)	$0.412 \pm 0.044$	$0.163 \pm 0.063$
MLP (Crop)	$0.374 \pm 0.022$	$0.086 \pm 0.029$

## 6 Visualização / Interação

Foi construída uma **demo interativa** em TensorFlow, utilizando o pipeline `tf.data` para carregamento otimizado, com as funções `map()`, `cache()` e `prefetch(AUTOTUNE)`.

## 7 Resultados

A CNN com Resize superou amplamente os demais modelos. A melhor configuração atingiu **Acurácia = 74.3% e F1 = 0.685** (Seed 44).

As curvas de treino/validação evidenciam maior estabilidade da CNN, enquanto o MLP sofreu forte underfitting.

## 8 Análise de Erros

A matriz de confusão revela que a CNN com Resize obteve melhor distinção entre classes, mas apresentou confusões em categorias visualmente similares (ex.: papel vs. papelão).

## 9 Conclusões e Próximos Passos

- CNN + Resize é o modelo vencedor, estatisticamente robusto e eficiente.
- MLP não é adequado para imagens complexas.
- Próximos passos:
  - Uso de Weighted Loss ou Focal Loss;
  - Data Augmentation focado em classes minoritárias;
  - Implementar Early Stopping no deploy.

## 10 Reprodutibilidade

- Código em Python/TensorFlow;
- Execução: comando único `python main.py`;
- Tempo total:  $\sim 2$  horas em GPU Tesla T4;
- Versões: TensorFlow 2.15, Scikit-learn 1.5.

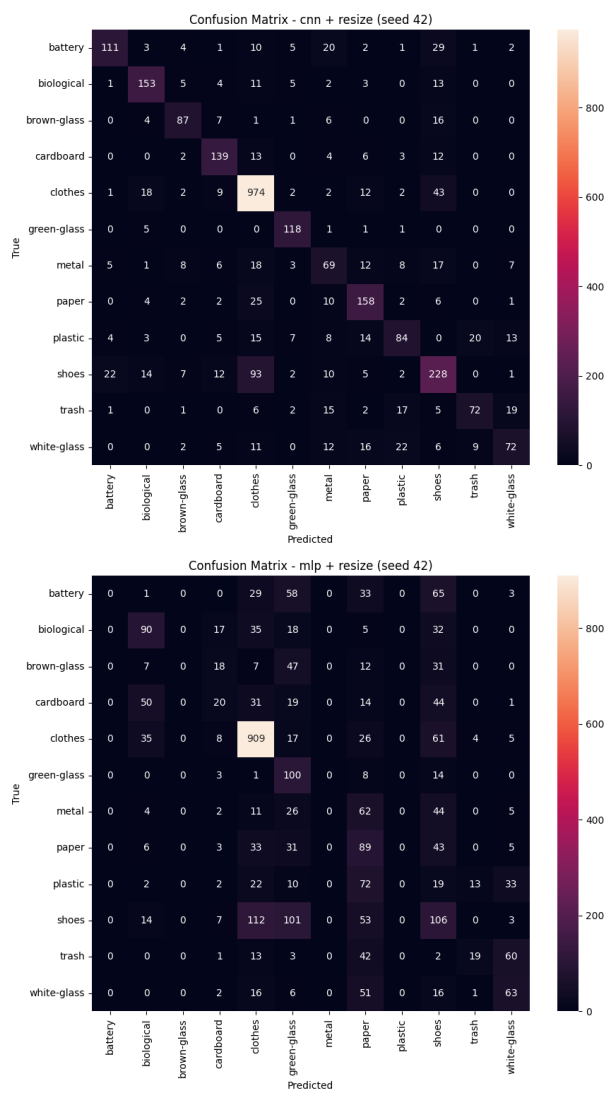


Figura 2: Matrizes de confusão: CNN (esq.) vs. MLP(dir.).

## Apêndice

```
model = Sequential([
    Conv2D(32, (3,3), activation='relu'),
    MaxPooling2D(),
    GlobalAveragePooling2D(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(12, activation='softmax')
])
```

Repositório com código completo: <https://github.com/oalmirsergio/midterm-nn-avaliacaol>