SPRINT 1 – Geração do Dataset e Treinamento do Modelo

Integrantes do grupo:

Danilo Ramalho Silva | RM: 555183

Israel Dalcin Alves Diniz | RM: 554668

João Vitor Pires da Silva | RM: 556213

Matheus Hungaro | RM: 555677

Pablo Menezes Barreto | RM: 556389

Tiago Toshio Kumagai Gibo | 556984

Link do colab: https://colab.research.google.com/drive/10-KGg39zyewytkkKhJiT_TPZ0pa1j2c9?usp=sharing

1. Montagem do Dataset

Fontes: as imagens foram coletadas manualmente no Mercado Livre, escolhendo anúncios de cartuchos HP originais e de cartuchos falsificados/outras marcas.

Critérios de seleção: preço médio do produto, avaliação do anúncio, tipo de loja (oficial ou não), comentários dos compradores.

Quantitativo: 60 imagens no total – 30 em dataset/HP_Original e 30 em dataset/Outros.

Pré-processamento:

- Redimensionamento para 224×224 px
- Normalização automática pelo image_dataset_from_directory (pixel values em [0,1])
- Divisão treino/validação 80/20

2. Estrutura da Rede Convolucional

2.1 Camadas

1. Conv2D(32, 3x3) → ReLU
2. MaxPooling2D
3. Conv2D(64, 3x3) → ReLU
4. MaxPooling2D
5. Conv2D(128, 3x3) → ReLU
6. MaxPooling2D
7. Flatten
8. Dense(64) → ReLU
9. Dense(2) → Softmax
10.

2.2 Razões para a estrutura utilizada

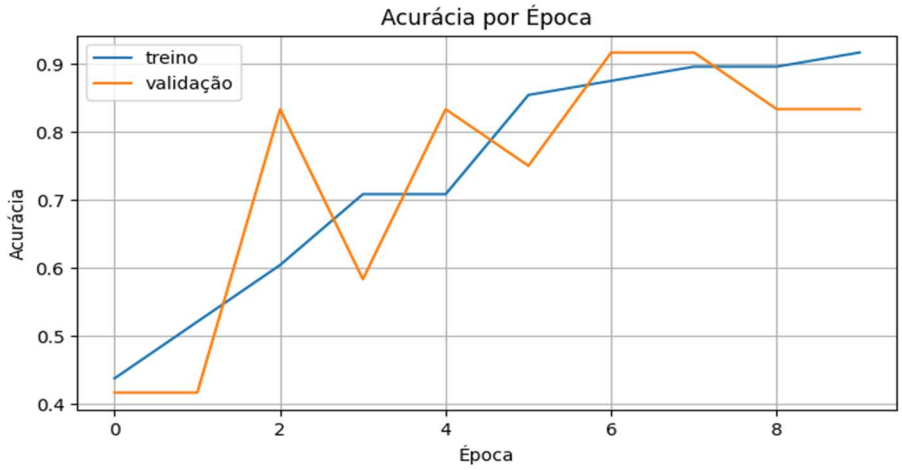
- **Complexidade Progressiva:** começa simples, aprendendo características básicas, evoluindo para camadas mais complexas.

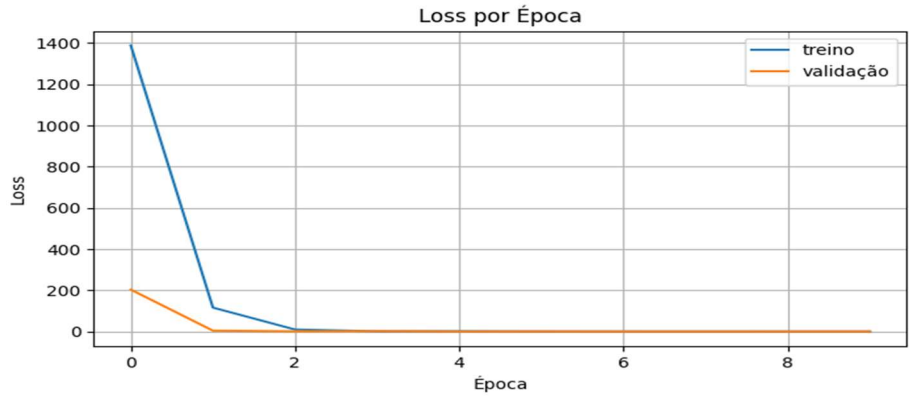
- **Controle de Overfitting:** intercala camadas convolucionais e MaxPooling para reduzir o volume de parâmetros e evitar sobreajuste.
- **Compatível com poucos dados:** ideal para datasets pequenos, como o descrito, mantendo um equilíbrio entre capacidade de aprendizado e generalização.
- **Desempenho bom com pouca profundidade:** estruturas menores, como está, são eficientes e rápidas de treinar, adequadas para modelos iniciais e experimentações rápidas.

Essa escolha estrutural facilita uma aprendizagem eficaz inicial e serve como ponto de partida para otimizações futuras

3. Gráficos e Interpretação

3.1 Curvas de Treino × Validação





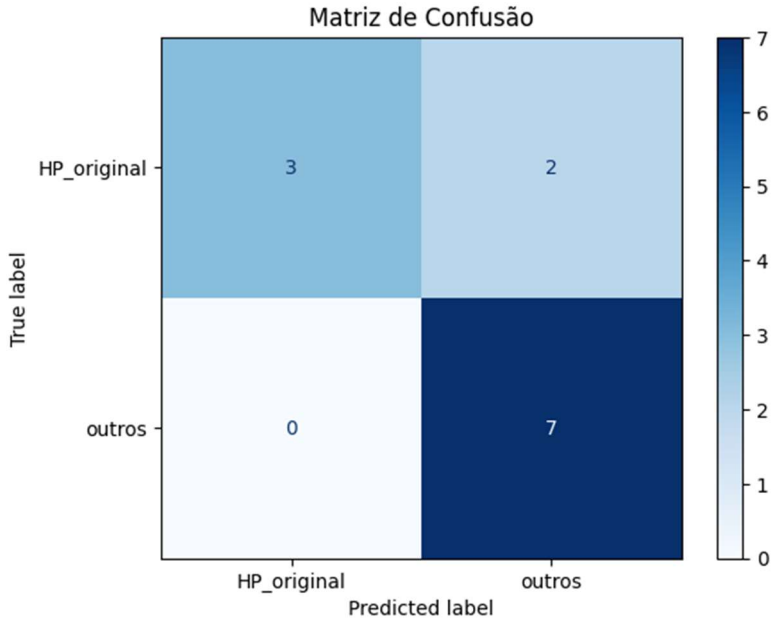
Loss: Treino cai de ≈1400 (época 0) para ≈10 (época 2) e chega quase a zero já na época 3.

Validação inicia em ≈200, acompanha a queda e estabiliza próximo de zero.

Acurácia: Treino sobe de 44% para 92% em 10 épocas.

Validação oscila: 42% (ép 0–1) → 83% (ép 2) → 58% (ép 3) → picos de 92% (ép 6–7) → encerra em ~83%.

3.2 Matriz de Confusão

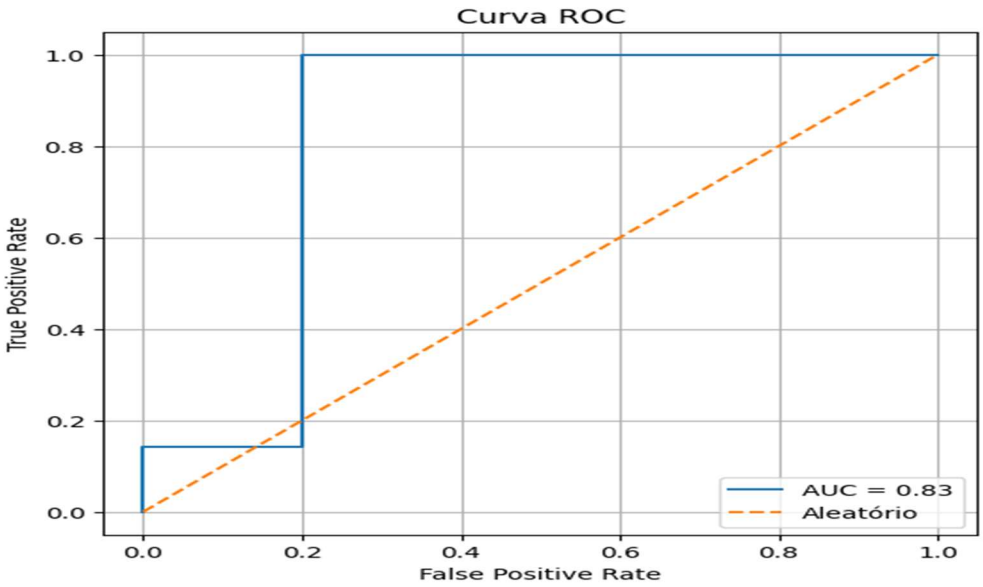


HP_original: recall 3/5 = 60%, precision 3/3 = 100%

outros: recall 7/7 = 100%, precision 7/9 ≈ 78%

Comentário: o modelo nunca classifica “outros” como “original” (nenhum falso positivo em HP_original), mas perde 40% dos originais.

3.3 Curva ROC

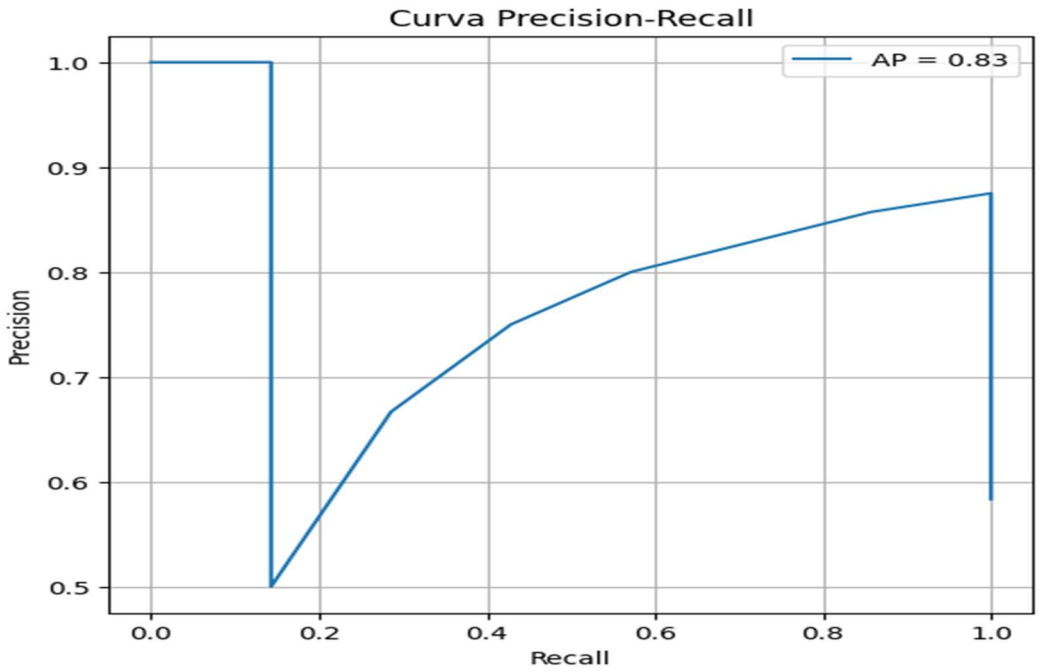


AUC = 0,83

A TPR atinge 100% já com FPR ≈20%.

Comentário: discriminação boa (AUC⍰0,5 = acaso), mas salto brusco reflete thresholds discretos e poucos dados.

3.4 Curva Precision-Recall

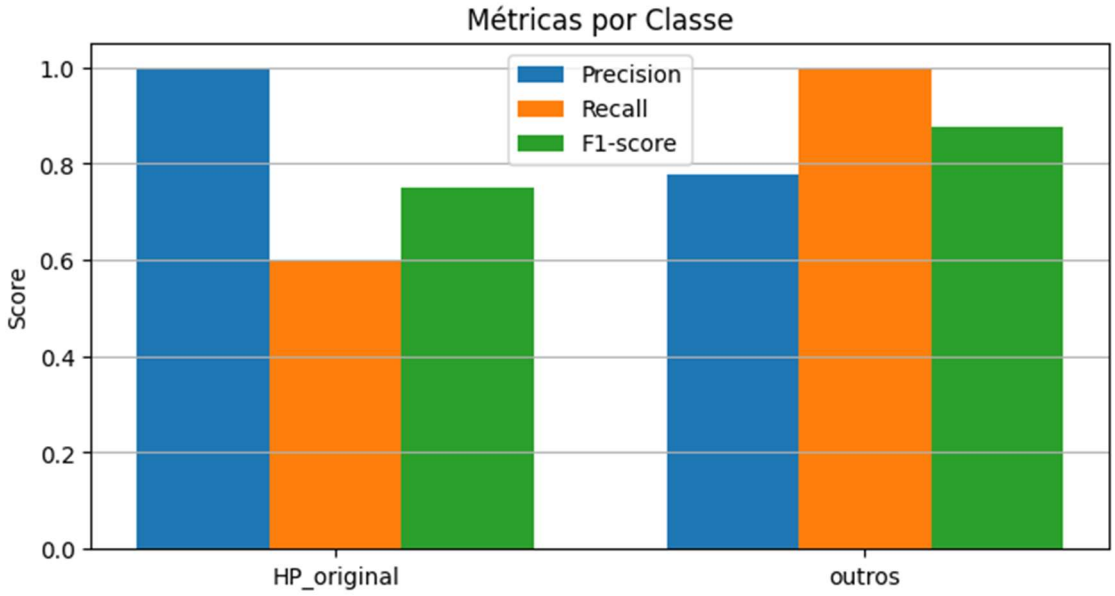


AP = 0,83

Precisão máxima (1,0) em recalls muito baixos; cai para 0,50 em recall ≈0,15; recupera-se para ~0,75–0,85 em recall 0,4–0,8; em recall=1,0, precisão ≈0,58.

Comentário: bom equilíbrio geral, mas exige escolher threshold intermediário (recall≈0,4–0,8) para manter precisão elevada sem descartar muitos positivos.

3.5 Métricas por Classe (Precision / Recall / F1-score)



Comentário: "HP_original" tem precisão perfeita mas recall baixo; "outros" recall perfeito mas aceita alguns originais como negativos.

4. Acurácia Final do Modelo

Validação: 83% (último valor de history.history['val_accuracy'])

5. Conclusão

O modelo simples aprendeu rapidamente a distinguir cartuchos originais vs. outros, atingindo 92% de treino e 83% de validação. Entretanto, a alta variância e os falsos negativos em “HP_original” (40%) indicam necessidade de:

- Aumento dos dados (especialmente para a classe original) ou coleta de mais imagens.
- Ajuste de limiar de decisão para equilibrar precisão e recall.
- Experimentar backbone pré-treinado leve (ex.: MobileNetV2) e comparar ganhos de performance.