

Modelo

Links Google Colab

Com Pré-Processamento: [Com Pré-Processamento](#)

Sem Pré-Processamento: [Sem Pré-Processamento](#)

Link Dataset Kaggle: Augmented Dataset for Fruits (Rotten/Fresh)

Link: [Augmented Dataset for Fruits \(Rotten/Fresh\)](#)

Por que essa arquitetura?

Escolhemos a combinação PCA + SVM porque o objetivo do projeto é demonstrar técnicas de visão computacional e aprendizado de máquina de forma compreensível, rápida e reproduzível, adequada para apresentação ao vivo.

- **O PCA (Principal Component Analysis)** reduz a dimensionalidade das imagens, preservando as principais características visuais, como cor, textura e padrões de deterioração, e facilita o treino do modelo.
- **O LinearSVC (SVM linear)** trabalha bem com dados de alta dimensionalidade e fornece decisões rápidas, mesmo com muitas imagens.
- Essa arquitetura é simples o suficiente para explicar passo a passo como o modelo aprende a diferenciar frutas frescas e podres, sem exigir GPU ou treinamento longo.

Como funciona a normalização?

A normalização é aplicada convertendo os valores dos pixels de **0–255** para **0–1**, usando:

- `img.astype("float32") / 255.0`

Isso padroniza a escala das entradas, o que ajuda o PCA e o SVM a convergirem de forma mais estável.

Sem normalização, os valores grandes dos pixels poderiam gerar distorções nos cálculos de distância do SVM e dificultar a separação entre classes.

Portanto, a normalização melhora precisão, velocidade de treinamento e estabilidade do modelo.

Por que o threshold 0?

No código usamos o `decision_function` do **LinearSVC**, que retorna um valor contínuo:

- **Valores > 0** → classifica como **Rotten** (podre)
- **Valores ≤ 0** → classifica como **Fresh** (fresco)

O ponto 0 funciona como **threshold neutro**.

- Se quisermos priorizar **segurança** (evitar consumo de fruta podre), podemos reduzir o threshold, aumentando a detecção de Rotten.
- Se quisermos priorizar **reduzir desperdício**, podemos aumentar o threshold, diminuindo a quantidade de Fresh classificada como Rotten.

Quais os limites do modelo?

Primeiramente, ele depende exclusivamente de sinais visuais. Alguns alimentos deteriorados não apresentam alterações externas no início, o que impede o modelo de identificar o problema.

Outro ponto é a variabilidade das condições do usuário: iluminação ruim, sombras, câmera fraca ou ângulos incomuns podem gerar erros.

Além disso, um dataset pequeno ou com poucas frutas estragadas pode gerar sobreajuste e reduzir a capacidade de generalização.

Por fim, alterações químicas e microbiológicas não são detectadas pelo modelo visual — nossa solução é auxiliar, não substituir inspeção humana adequada.