**Descrição do problema**

O código tem como objetivos aplicar processamentos de imagens e classificação com IA. O processamento ocorre na classe processamento.py, ela usa as técnicas de redimensionamento para (128x128), filtro Gaussiano (cv2.GaussianBlur) e equalização de histograma (cv2.equalizeHist, se em tons de cinza).

**Justificativa das técnicas realizadas**

Redimensionamento (cv2.resize)

Objetivo: Padronizar o tamanho das imagens para facilitar comparações, visualizações e uso em modelos de aprendizado de máquina.

Justificativa: Garante uniformidade no conjunto de dados e reduz a complexidade computacional.

Conversão para Escala de Cinza (cv2.cvtColor com cv2.COLOR\_BGR2GRAY)

Objetivo: Reduzir a imagem de 3 canais (cor) para 1 canal (intensidade).

Justificativa: Remove informações de cor desnecessárias, reduz o custo computacional e mantém as informações estruturais essenciais.

Desfoque Gaussiano (cv2.GaussianBlur)

Objetivo: Suavizar a imagem e eliminar ruídos de alta frequência.

Justificativa: Reduz detalhes irrelevantes que poderiam interferir em análises posteriores (como detecção de bordas ou segmentação).

Equalização do Histograma (cv2.equalizeHist)

Objetivo: Melhorar o contraste da imagem em tons de cinza.

Justificativa: Redistribui os níveis de intensidade para destacar regiões escuras ou claras, facilitando a extração de características importantes.

Etapas realizadas

Leitura da Imagem

A imagem é carregada a partir do caminho indicado usando cv2.imread.

Essa é a etapa inicial para obter os dados brutos da imagem no formato manipulável (matriz NumPy).

**Etapas realizadas**

**Redimensionamento**

A imagem é redimensionada para 128x128 pixels com cv2.resize.

**Conversão para Escala de Cinza**

A imagem original é convertida para tons de cinza usando cv2.cvtColor.

A imagem em escala de cinza mantém apenas a intensidade de luz, descartando as cores, o que simplifica o processamento.

**Aplicação do Desfoque Gaussiano**

É aplicado um filtro de suavização com cv2.GaussianBlur.

Essa etapa remove ruídos finos da imagem, suavizando as variações bruscas

**Equalização do Histograma**

A imagem em tons de cinza passa por cv2.equalizeHist.

Essa etapa melhora o contraste global da imagem, realçando regiões com pouca variação de intensidade.

Visualização das Etapas

Cada imagem resultante das etapas acima é exibida em uma única figura com 5 colunas, usando matplotlib.pyplot.

A visualização permite comparar visualmente as transformações aplicadas em cada fase do processamento.

**Resultados obtidos**

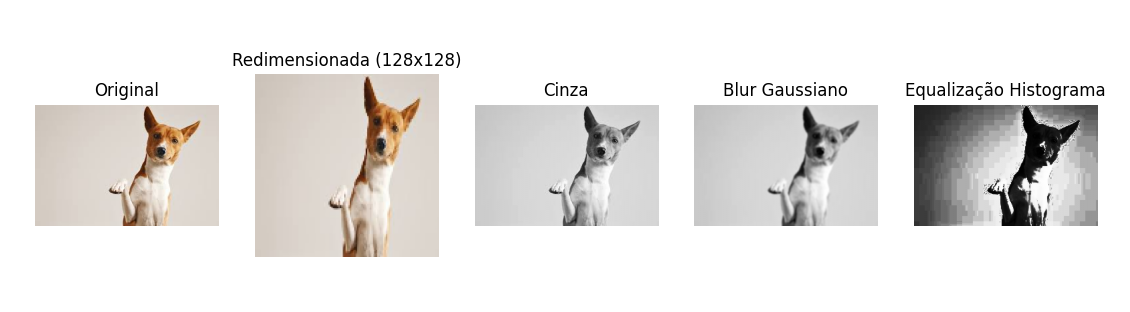
Cachorro1.pgj

****

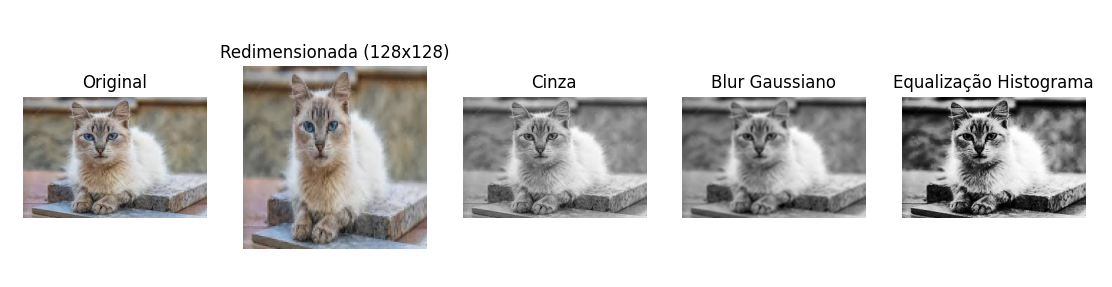
Cachorro2.pgj



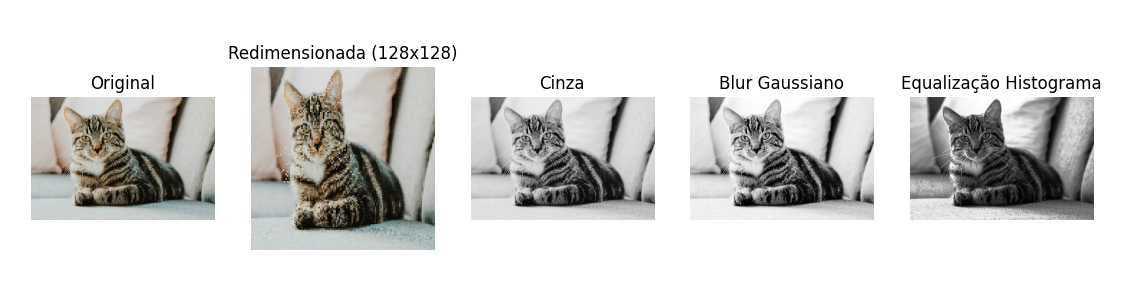
Cachorro3.jpg



Gato1.jpg

****

Gato2.jpg



Gato3.jpg



**Tempo total gasto**

2 horas para realizar o que foi entregue

**Dificuldades encontradas**

A maior dificuldade encontrada foi conseguir processar todas as imagens sem precisar carregar uma por uma de forma manual e encontrar uma forma adequada para a visualização dos resultados