VISÃO COMPUTACIONAL 2024

Cerebroto

Especificação Técnica

Objetivo | Classificação e técnicas | Número de classes | Documentação do código | Resultados





Objetivo

O objetivo deste projeto é desenvolver e treinar um modelo capaz de classificar imagens de vegetais em diferentes categorias, utilizando técnicas de visão computacional e aprendizado profundo.

Classificação e técnicas

O problema abordado é de classificação multi-classe, no qual cada imagem de vegetal é atribuída a uma das várias categorias predefinidas. O projeto emprega técnicas de aprendizado profundo, especificamente a arquitetura de Rede Neural Convolucional (CNN), para extrair características das imagens e realizar a classificação.

Número de classes

O número de classes é determinado pela variável num_classes, configurada como 15 na função construir_modelo(). Portanto, o modelo está projetado para classificar as imagens em 15 categorias distintas de vegetais, sendo eles Feijão, cabaça amarga, cabaça, berinjela, brócolis, repolho, pimentão, cenoura, couve-flor, pepino, mamão, batata, abóbora, rabanete, tomate. Vale lembrar que o modelo classifica estes vegetais em inglês.

Documentação do código

Constructor_classificador_de_vegetais

Preparação do Ambiente

As bibliotecas essenciais são importadas e o Google Drive é montado para acessar os dados de treinamento, teste e validação armazenados.

numpy: Manipulação eficiente de arrays e operações matemáticas.

pandas: Estruturas de dados para manipulação de tabelas e séries temporais.

pathlib: Manipulação de caminhos de arquivos de maneira orientada a objetos.

os.path: Operações comuns com caminhos de arquivos e diretórios.

matplotlib.pyplot: Visualização de dados por meio de gráficos e imagens.

tensorflow: Framework de aprendizado profundo utilizado para construção e treinamento de modelos de rede neural.

google.colab.drive: Monta o Google Drive no Colab para acesso a arquivos armazenados na nuvem.

pprint: Impressão de estruturas de dados em um formato legível.

seaborn: Visualização de dados baseada em Matplotlib para criar gráficos estatísticos.

sklearn.metrics: Avaliação de modelos de machine learning usando métricas como acurácia e matriz de confusão.

PIL (Python Imaging Library): Processamento e manipulação de imagens.

IO: Manipulação de streams de entrada e saída, especialmente útil para tratar imagens em bytes.

Coleta de Dados

Utilizando o Path do módulo pathlib, são gerados caminhos para os arquivos de imagem em três diretórios: treinamento, teste e validação. Esses caminhos são convertidos em listas.

Processamento de Imagens: A função processar_imagens é criada para extrair rótulos dos caminhos das imagens e organizar essas informações em DataFrames do Pandas, que são embaralhados para garantir uma distribuição aleatória.

Visualização de Dados

Alguns exemplos de imagens e rótulos são exibidos usando matplotlib para garantir a correta leitura e processamento dos dados.

Geradores de Imagem: Os ImageDataGenerators são configurados para aplicar pré-processamento adequado, incluindo a normalização das imagens usando a função de pré-processamento do MobileNetV2. Aumento de dados é aplicada ao conjunto de treinamento para melhorar a robustez do modelo.

Construção do Modelo

Um modelo baseado no MobileNetV2 pré-treinado é definido. A arquitetura é estendida com camadas densas adicionais, finalizando com uma camada softmax para classificação em 15 classes. O modelo é compilado com o otimizador Adam e a função de perda categórica.

Treinamento do Modelo

O modelo é treinado usando o gerador de imagens de treinamento, com validação simultânea. Callbacks são usados para evitar overfitting, interrompendo o treinamento se a perda de validação não melhorar após duas épocas consecutivas.

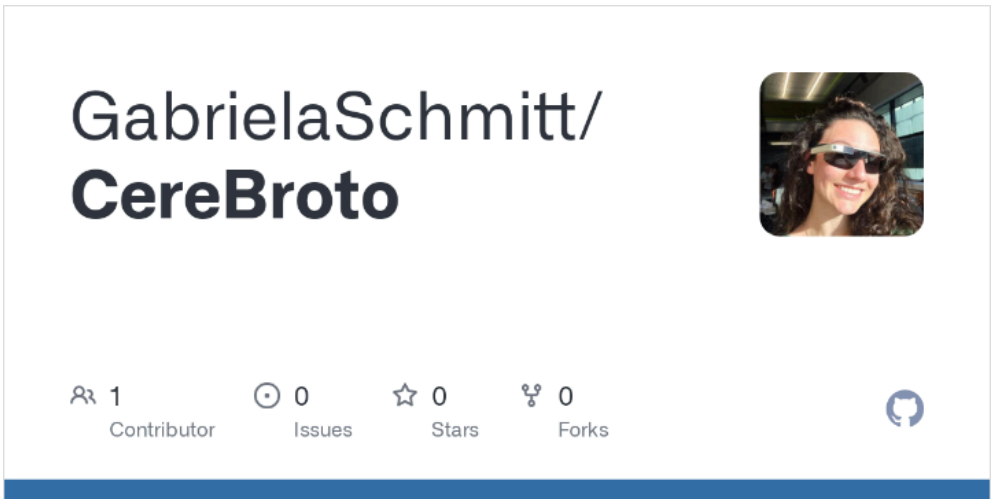
Avaliação e Visualização de Resultados: As métricas de acurácia e perda durante o treinamento são plotadas. A performance no conjunto de teste é avaliada e a matriz de confusão é exibida usando seaborn para visualizar a precisão por classe.

Grad-CAM

Funções adicionais para geração de mapas de calor Grad-CAM são fornecidas para interpretar visualmente quais partes das imagens influenciam mais as decisões do modelo.

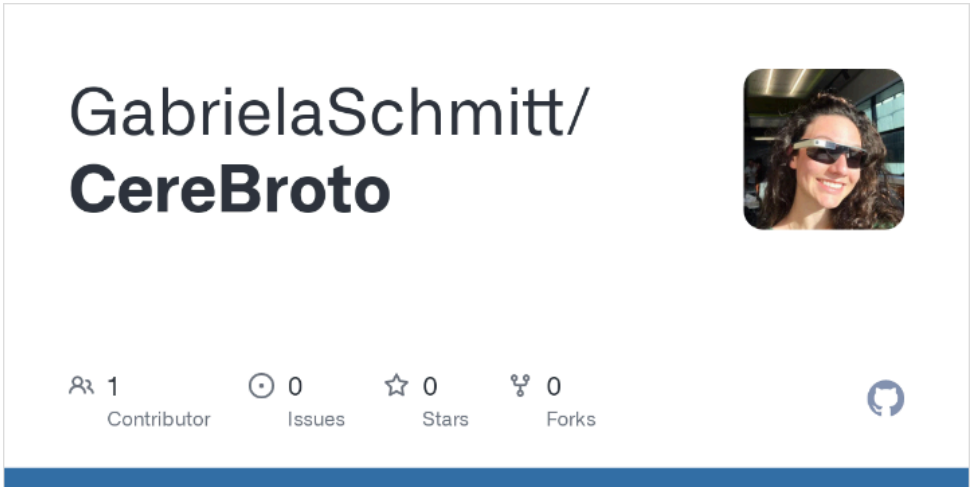
Download do Modelo

O modelo treinado é salvo em um arquivo .h5 para uso posterior.



Inferência_Vegetais

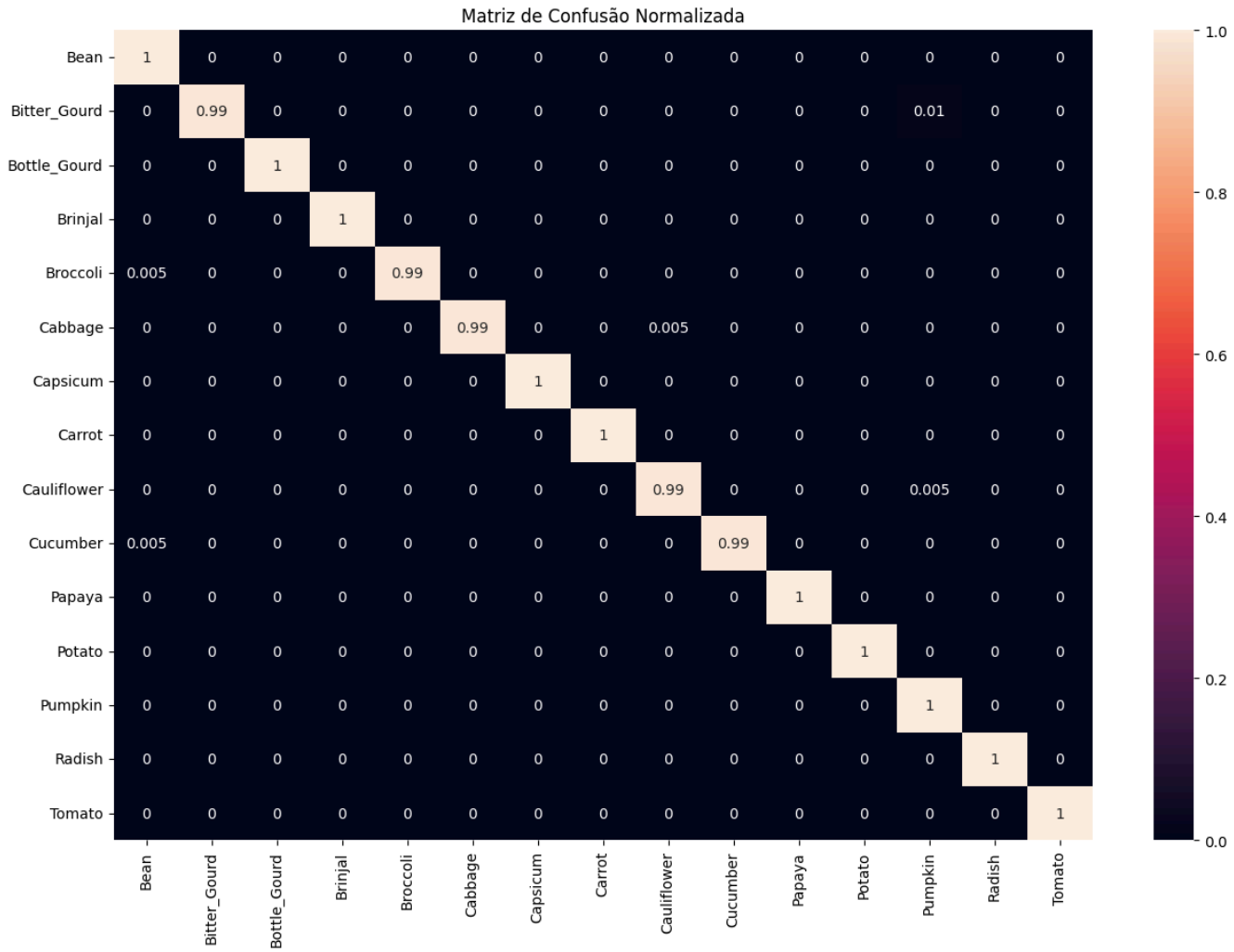
Este código Python implementa uma interface interativa no Google Colab para carregar e classificar imagens de vegetais usando o modelo Keras previamente treinado. O script inicia instalando as bibliotecas necessárias (ipywidgets, Pillow, tensorflow). Em seguida, importa os módulos essenciais e carrega o modelo salvo em formato .h5 do Google Drive. Define-se um dicionário com as classes de vegetais para mapeamento dos resultados do modelo. Utiliza-se ipywidgets para criar um botão de upload que permite ao usuário carregar uma imagem. A função on_image_upload processa a imagem carregada: converte para o formato RGB, redimensiona para 224x224 pixels, normaliza os valores dos pixels e transforma a imagem em um array adequado para o modelo. Após o pré-processamento, a imagem é passada pelo modelo para inferência, e o rótulo predito é exibido. A função é associada ao widget de upload, garantindo que a classificação ocorra automaticamente após o upload da imagem.



Resultados

O classificador de vegetais por imagem obteve um resultado excelente, com **99,8% de precisão** no conjunto de teste. Isso significa que, para cada 100 imagens de vegetais avaliadas, o modelo classificou 99 corretamente.

Análise da Matriz de Confusão



Matriz de Confusão: A imagem apresenta uma matriz de confusão normalizada, que ajuda a visualizar como o modelo se comportou em cada classe de vegetais.

Classes: As 15 classes de vegetais utilizadas no treinamento do modelo estão representadas nas linhas e colunas da matriz.

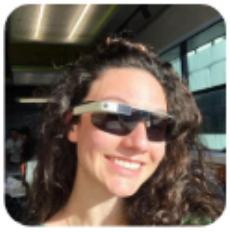
Cor: A intensidade da cor em cada célula da matriz indica a proporção de imagens de uma classe específica que foram erroneamente classificadas como outra classe. Cores mais claras (próximas ao branco) indicam maior concordância entre a classe real (linhas) e a classe prevista (colunas). Cores mais escuras indicam mais discordância.

Valores: Os números dentro das células representam a quantidade real de imagens que se encaixam em cada categoria. Por exemplo, o valor na linha "Feijão" e coluna "Feijão" é 0, o que significa que todas as imagens de feijão foram classificadas corretamente. Por outro lado, o valor na linha "Pepino" e coluna "Brócolis" é 0,005, indicando que uma pequena fração de imagens de pepino foi mal classificada como brócolis.

O alto nível de precisão (99,8%) e a predominância de cores claras na matriz de confusão demonstram que o classificador de vegetais por imagem é altamente confiável e eficiente na identificação de diferentes tipos de vegetais a partir de imagens.

Acesse nosso repositório do GITHUB

GabrielaSchmitt/
CereBroto



1 Contributor | 0 Issues | 0 Stars | 0 Forks