**Detecção de Câncer de Pele usando Aprendizado de Máquina**

Este projeto visa a detecção de câncer de pele por meio de técnicas de aprendizado de máquina. O modelo desenvolvido é uma Rede Neural Convolucional (CNN) implementada com TensorFlow e Keras, feito no Google Colab. O conjunto de dados utilizado para treinamento e teste é proveniente do dataset [Skin Cancer MNIST: HAM10000](https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000/data), que inclui imagens de várias lesões de pele.

**Instruções de Uso**

Requisitos e Dependências

* Python
* TensorFlow
* Keras
* NumPy
* pandas
* matplotlib
* OpenCV
* scikit-learn

Configuração

1. Clone ou faça o download do repositório do projeto para o seu computador local.
2. Instale as dependências necessárias usando o seguinte comando:

*pip install tensorflow keras numpy pandas matplotlib opencv-python scikit-learn*

Obs.: Utilizar o Anaconda ou o Google Colab facilita o uso das dependências.

Conjunto de Dados

O conjunto de dados dataset-skincancer é necessário para treinamento e teste. É importante importar o arquivo zipado dentro da pasta do projeto e rodar a primeira linha de código para descompactá-lo.

Executando o Código

1. Abra o ambiente de preferência onde o código está salvo.
2. Rode a primeira linha de comando isolada para descompactar o arquivo de dados. Sem esses arquivos de dados, **não é possível** dar sequência na execução projeto.
3. Execute cada célula de código sequencialmente para carregar o conjunto de dados, pré-processar as imagens, criar e treinar o modelo, e avaliar seu desempenho.

Testando com uma Nova Imagem

1. Certifique-se de que o modelo está salvo como "modelo.keras" após o treinamento.
2. Modifique a variável *image\_path* na última seção para o caminho da imagem que você deseja testar.
3. Execute a última seção para exibir a imagem e prever sua classe.

**Arquitetura do Modelo**

A CNN implementada consiste em camadas convolucionais, camadas de max-pooling, camadas de dropout e camadas totalmente conectadas. O modelo é compilado com o otimizador Adam e a perda de entropia cruzada categórica.

O modelo é dividido em 80% para treinamento e 20% para testes. Seu treino acontece para 50 épocas, 20% de dados para validação e tamanho do batch em 92

**Resultados**

Após o treinamento, as métricas de desempenho do modelo, como precisão e perda, são visualizadas em gráficos. Além disso, o modelo é testado em um conjunto separado de imagens, e o resultado da precisão é impresso.