

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e Informática

Pedro Henrique Ferreira de Souza

**Processamento de Imagens:**

Reconhecimento de padrões por textura em imagens mamográficas

Belo Horizonte

2022

Processamento de Imagens

Alexei Manso Correa Machado

Trabalho prático com o intuito de praticar os

conhecimentos apresentados e adquiridos

durante as aulas da disciplina de

Processamento de Imagens, do curso de

Ciência de Computação da Pontifícia

Universidade Católica de Minas Gerais.

Belo Horizonte

2022

**Sumário**

1. **INTRODUÇÃO**
   1. Objetivos e Informações gerais
2. **IMPLEMENTAÇÃO**
   1. Ambiente de desenvolvimento
   2. Bibliotecas utilizadas
3. **DESCRIÇÃO SOBRE IMPLEMENTAÇÃO**
   1. Ler e visualizar imagens
   2. Quantização
   3. Haralick
   4. Rede neural
      1. Treinar os classificadores
      2. Matriz de confusão
   5. Classificar imagem
   6. Especificidade e acurácia
4. **TESTES**
   1. Métodos do Haralick
      1. Entropia
      2. Homogeneidade
      3. Energia
   2. Classificador
      1. Acurácia
      2. Especificidade
      3. Matriz de confusão
   3. Análise do corte
      1. Classe de predição
   4. Tempo de execução
   5. Teste da Rede Neural
5. **CONCLUSÃO**
6. **BIBLIOGRAFIA**
7. **INTRODUÇÃO**

O desenvolvimento das técnicas de processamento de imagens digitais ocorridos nas últimas décadas propiciou o surgimento de uma grande variedade de aplicações em diversas áreas e aliado a expansão do aprendizado de máquina, em especial às Redes Neurais.

O objetivo do trabalho era construir um software para reconhecimento de padrões por textura em imagens de mamografia utilizando conceitos obtidos na disciplina Processamento de Imagens.

Foi construído um software na linguagem Python 3 capaz de abrir imagens, alterar a quantidade de tons de cinza dessas imagens e classifica-las após um treino da rede neural implementada.

Usando descritores de Haralick, foram treinadas centenas de imagens que são separadas em quatro classes diferentes de acordo com a escala de densidade BIRADS. Esse treinamento servirá para classificarmos imagens de *input* do usuário.

1. **IMPLEMENTAÇÃO**
   1. **Ambiente de desenvolvimento**

O programa foi desenvolvido em um sistema operacional Windows 11 utilizando a linguagem Python 3 (versão 3.8.5).

Para executar o código é necessária a instalação do programa Python, encontrado no site <https://www.python.org/>. É sempre bom lembrar que se a versão do Python for diferente da 3.8.5 pode haver alguma incompatibilidade com as bibliotecas usadas.

* 1. **Bibliotecas utilizadas**

Foram utilizadas as seguintes bibliotecas e suas respectivas versões:

* Matplotlib (3.5.1)

- plotagem de gráficos

* Numpy (1.21.1)

- processamento numérico

* Opencv-python (4.5.3)

- visão computacional (captura e saída de imagem)

* Mahotas (1.4.12)

- processamento de imagens

* Scikit-learn (0.24.2)

- classe e funções para Rede Neural

* Pillow (8.3.1)

- tratamento de imagens

* Scikit-image (0.17.2)

- coleção de algoritmos para processamento de imagens

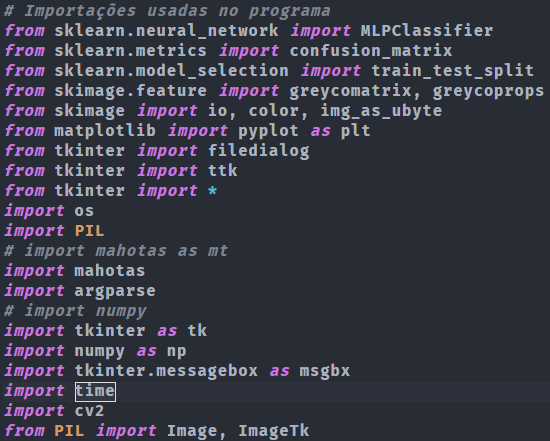
Para instalar as bibliotecas utiliza-se o gerenciador de pacotes do Python, PIP.

* pip install \*nome\_da\_biblioteca\*

Para especificar a versão desejada:

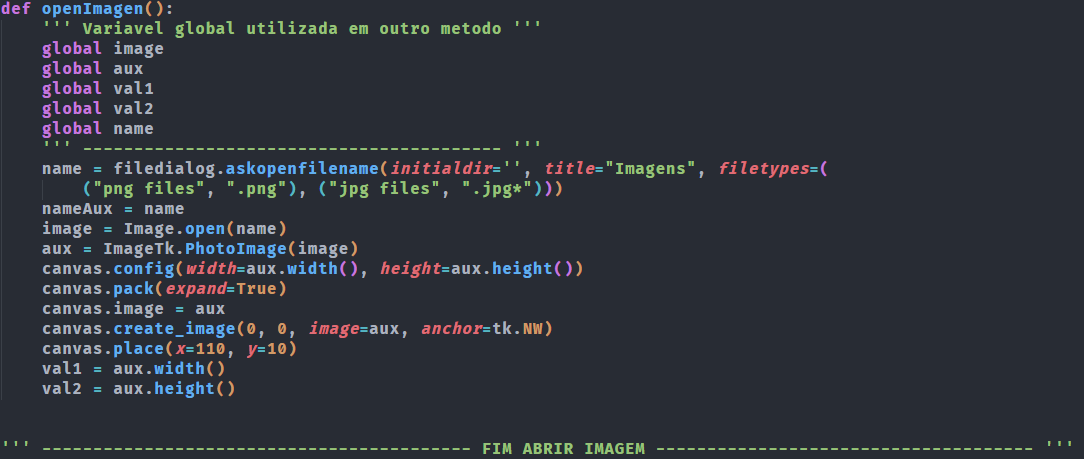
* pip install \*nome\_da\_biblioteca\*==x.x.x

Ex.: pip install mahotas==1.4.12



1. **DESCRIÇÃO SOBRE IMPLEMENTAÇÃO**
   1. **Ler e visualizar imagens**

Para abrir uma imagem do usuário, o método abaixo abre uma caixa de diálogo do sistema para escolher a imagem desejada. As extensões suportadas são ‘.png’ e ‘.jpg’. Após selecionar a imagem e confirmar, ela é carregada na interface do programa.



* 1. **Quantização**

O programa tem a possibilidade de selecionar a escala de tons de cinza que será aplicada na imagem. Foram implementadas duas escalas, 16 e 32 tons. Após selecionado o número de tons, a imagem na janela do programa é alterada e a nova imagem salva no diretório de execução é atualizada.

* 1. **Haralick**

Haralick é um dos descritores para classificação das imagens. Foi implementada uma função para calcular as propriedades de Haralick para matrizes de coocorrência circulares com distâncias iguais a 1, 2, 4, 8 e 16. O método recebe uma imagem e parâmetros com as características que serão analisadas. Nesse método, é retornada a matriz de coocorrência.

* 1. **Rede Neural**

Utilizando a classe MLP Classifier da biblioteca Sklearn, foi construída uma rede neural, automatizando a rede neural multicamada. Depois que a rede neural é treinada, é mostrada a qual classe de predição pertence a imagem selecionada pelo usuário.

O selecionador utilizado foi “***solver= ’adam’***” pelo fato de ser o padrão da rede neural. A quantidade de *neurônios* configurados foi ***hidden\_layer\_sizes = (200,200,200)***. Após isso, o classificador foi preparado com as quatrocentas imagens recebidas e com a matriz da imagem gerada pelo Haralick.

Depois de treinada, 25 imagens para cada classe serão testadas utilizando o método ‘***predict***’ que retorna os valores da classificação das imagens de teste. Finalizados os testes, é gerada a matriz de confusão com o método ‘***confusion-matrix***’.

* + 1. **Treinar os classificadores**

Ao clicar para treinar a rede neural, o usuário deve selecionar a pasta desejada onde estão os quatro diretórios das imagens de treino, numerados de 1 a 4.

Ao carregar as imagens, a rede neural começa a ser treinada.

* + 1. **Matriz de confusão**

Depois do treino da rede neural, a matriz de confusão é gerada. Usando-a, é possível calcular a acurácia e a especificidade do classificador.

* 1. **Classificar Imagem**

Tendo realizado o treino da rede neural, verificamos se o classificador existe. Caso positivo, é realizada a predição da imagem selecionada pelo usuário e então será mostrada a classe de predição que ela pertence.

* 1. **Especificidade e Acurácia**

Será calculada a acurácia e a especificidade visto que temos a matriz de confusão gerada.

O método para calcular a acurácia recebe a matriz de confusão, faz o cálculo da diagonal principal da matriz e divide para a quantidade de imagens de teste, sendo cem (25 para cada classe).

O método para o cálculo da especificidade usa os valores restantes e divide pela quantidade de imagens usadas no treino inicial, nesse caso, trezentas (75 para cada classe).

1. **TESTES**
   1. **Métodos de Haralick**

Primeiro passo a ser tomado é averiguar se os métodos de Haralick estavam gerando as matrizes de coocorrência.

* + 1. **Energia**