

Trabalho de Implementação

Gabriel Bicalho Ferreira 18.1.4024

Parte 1

A seguinte função, resolve as questões 1, 2, 3 e 4

```
function [matrizDados, etiqueta] = gerarImagensCalcularHuMoments(endereco)
    % função para os exercicios 1, 2, 3 e 4
    %endereco = 'C:\Users\gabri\Desktop\6 periodo\PDI\Lista 6\numbers\';
    matrizDados = zeros(7000, 7);
    etiqueta = zeros(7000, 1);
    elem = 1;
    ext = 'jpg';
    arquivos = dir(fullfile(endereco, [ '*' ext]));
    quant = length(arquivos);
```

Nessa parte, temos apenas algumas variáveis que serão necessárias para a função gerarImagensCalcularHuMoments.

Essa função irá gerar 7000 imagens e calcular o Hu Moments de cada uma das imagens, que é feito pelos quatro for aninhado demonstrados abaixo:

```
for i = 1:quant % laço para cada imagem
    img = imread( fullfile(endereco, arquivos(i,1).name));
    GR = 0; % grau de rotação inicial
    for j = 1: 20 % 19 rotações mais a imagem original
        nimg = imrotate(img, GR);
        GR = 1+round(rand(1)*359); %rotação em um angulo de 1 a 359
        escala = 0.5;
        for e = 1: 7 % escala
            nimg2 = imresize(nimg, escala);
            escala = escala + 0.25;
            ruído = 0.0;
            for r = 1:5 % ruídos
                nimg3 = imnoise(nimg2, "salt & pepper", ruído);
                vet = humoments(nimg3);
                matrizDados(elem,:) = vet;
                etiqueta(elem, 1) = i-1;
                elem = elem +1;
                ruído = ruído + 0.01;
            end
        end
    end
end
```

O primeiro for é para a leitura das imagens.

O segundo for é para realizar as rotações.

O terceiro for é para realizar a escala.

E por último, o quarto for é para adicionar o ruído e calcular o Hu moments de cada imagem gerada.

Nessa função, temos duas variáveis de retorno, primeiro é a matriz de dados gerados pelo cálculo do Hu moments. A segunda variável retornada seria a etiqueta de cada imagem.

Parte 2

Agora que já temos um conjunto expressivo de imagens e seus respectivos cálculos de Hu moments, podemos realizar um reconhecimento e padrões em imagens. Para isso foi dividido o conjunto em treino e teste.

A seguinte função, resolve o que foi pedido nos exercícios 5, 6 e 7.

```
function model = classif(endereco)
    [dados, etiqueta] = gerarImagensCalcularHuMoments(endereco);
    [train, test] = crossvalind('HoldOut', etiqueta, 0.25);
    train_data = dados(train, :); %dados de treino
    train_label = etiqueta(train); %etiquetas de treino
    test_data = dados(test, :);
    test_label = etiqueta(test);
    model = fitcdiscr(train_data, train_label); %modelo para treino
    pred = predict(model, test_data); % etiquetas de teste
    [lin, ~] = size(pred);
    cont = 0;
    for i = 1:lin
        if pred(i) == test_label(i)
            cont = cont + 1;
        end
    end
    tex = 'Porcentagem de acerto: ';
    disp(tex);
    disp(cont/lin);

    matrizConfusao = confusionmat(test_label, pred )
```

Essa função, utiliza a função do exercício anterior, e sobre o mesmo conjunto de imagens, gera um modelo para reconhecimento.

Os treinos foram feitos sobre 75% dos dados, os outros 25% dos dados foram utilizados para testes.

Como resultado, foi obtido um acerto aproximadamente 68%.

Abaixo, temos a imagem do resultado da execução, em que temos também a matriz confusão.

```
>> classif(dados, etiqueta);
Porcentagem de acerto:
    0.6806

>> classif(endereco);
Porcentagem de acerto:
    0.6829

matrizConfusao =

    87     0     0     0     0     0     3     0    77     8
     0   160     9     0     0     0     0     6     0     0
     0     2   161     0     0    12     0     0     0     0
     0     0     0   125     0    50     0     0     0     0
    34     0     0     1   134     0     4     0     2     0
     0     0    10    33     0   132     0     0     0     0
     5     0     0     5     2     0    41     0    60    62
     0     1     1     1     0     0     0   172     0     0
    55     0     0     0     1     0     0     0   117     2
     6     0     0     2     0     0    35     0    66    66
```

Parte 3

Na parte 3, será resolvido o exercício 8. Para isso, a intenção é utilizar o modelo já treinado na questão anterior, para resolver um problema específico.

```
function acertos = testNumber(endereco, img)
    model = classif(endereco);
    figs = BoundingBoxPatches(img);
    [lin, ~] = size(figs);
    dados = zeros(lin, 7);
    etiqueta = [0; 1; 3; 5; 2; 7];
    for i = 1:lin
        dados(i, :) = humoments(figs{i});
    end
    pred = predict(model, dados);

    acertos = 0;
    for i = 1:lin
        if pred(i) == etiqueta(i)
            acertos = acertos + 1;
        end
    end
    A = reshape(pred, [1,6]);
    B = reshape(etiqueta, [1,6]);
    disp(A);
    disp(B);
    disp(acertos);
end
```

Na imagem abaixo, temos que o primeiro vetor são as etiquetas geradas, o segundo vetor são as etiquetas corretas e o terceiro valor é a quantidade de acertos.

8	1	3	3	2	7
0	1	3	5	2	7
4					

Dos 6 números, o algoritmo conseguiu identificar 4.