



UFOP

Universidade Federal
de Ouro Preto

Lista 05

**Aluno em Graduação da Universidade
Federal de Ouro Preto do curso Ciência da**

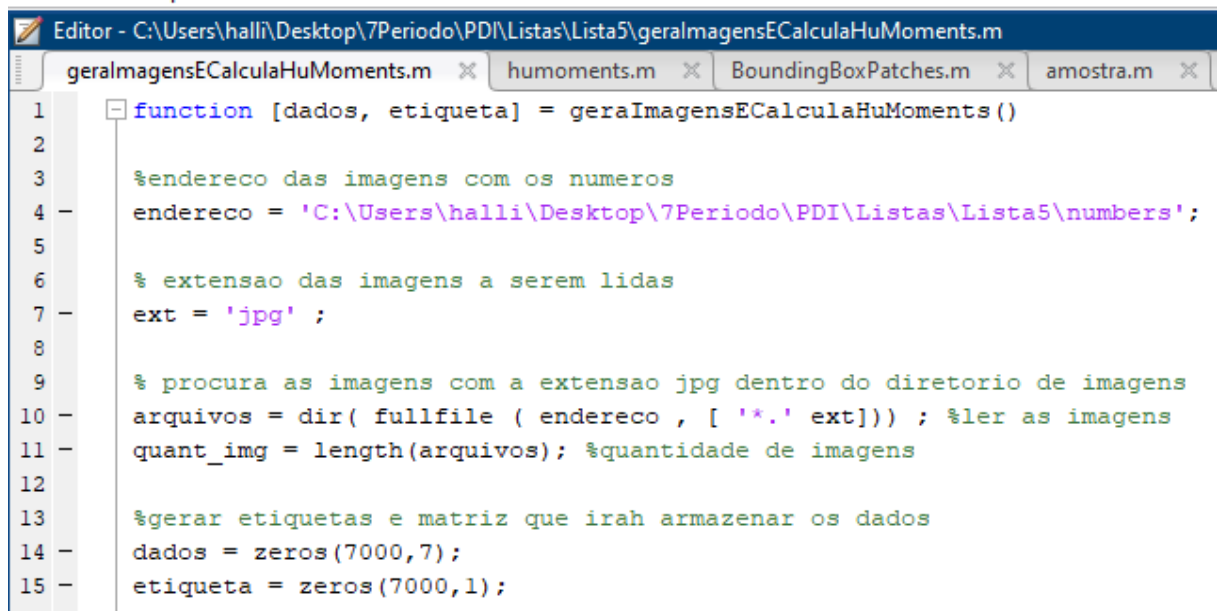
Computação:

Halliday Gauss Costa dos Santos.

Matrícula: 18.1.4093.

Área: Processamento de Imagens.

Questões 1, 2, 3 e 4)



```
Editor - C:\Users\halli\Desktop\7Periodo\PDI\Listas\Lista5\geraImagensECalculaHuMoments.m
geraImagensECalculaHuMoments.m  X  humoments.m  X  BoundingBoxPatches.m  X  amostra.m  X

1  function [dados, etiqueta] = geraImagensECalculaHuMoments()
2
3      %endereço das imagens com os numeros
4  -  endereco = 'C:\Users\halli\Desktop\7Periodo\PDI\Listas\Lista5\numbers';
5
6      % extensao das imagens a serem lidas
7  -  ext = 'jpg' ;
8
9      % procura as imagens com a extensao jpg dentro do diretorio de imagens
10 -  arquivos = dir( fullfile ( endereco , [ '*' ext] ) ) ; %ler as imagens
11 -  quant_img = length(arquivos); %quantidade de imagens
12
13      %gerar etiquetas e matriz que irah armazenar os dados
14 -  dados = zeros(7000,7);
15 -  etiqueta = zeros(7000,1);
```

A função 'geraImagensECalculaHuMoments' representada acima faz a leitura de todo um diretório procurando imagens de arquivo 'jpg'. Esse diretório acima, chamado de 'numbers', só possui imagens dos números de 0 a 9 (uma de cada).

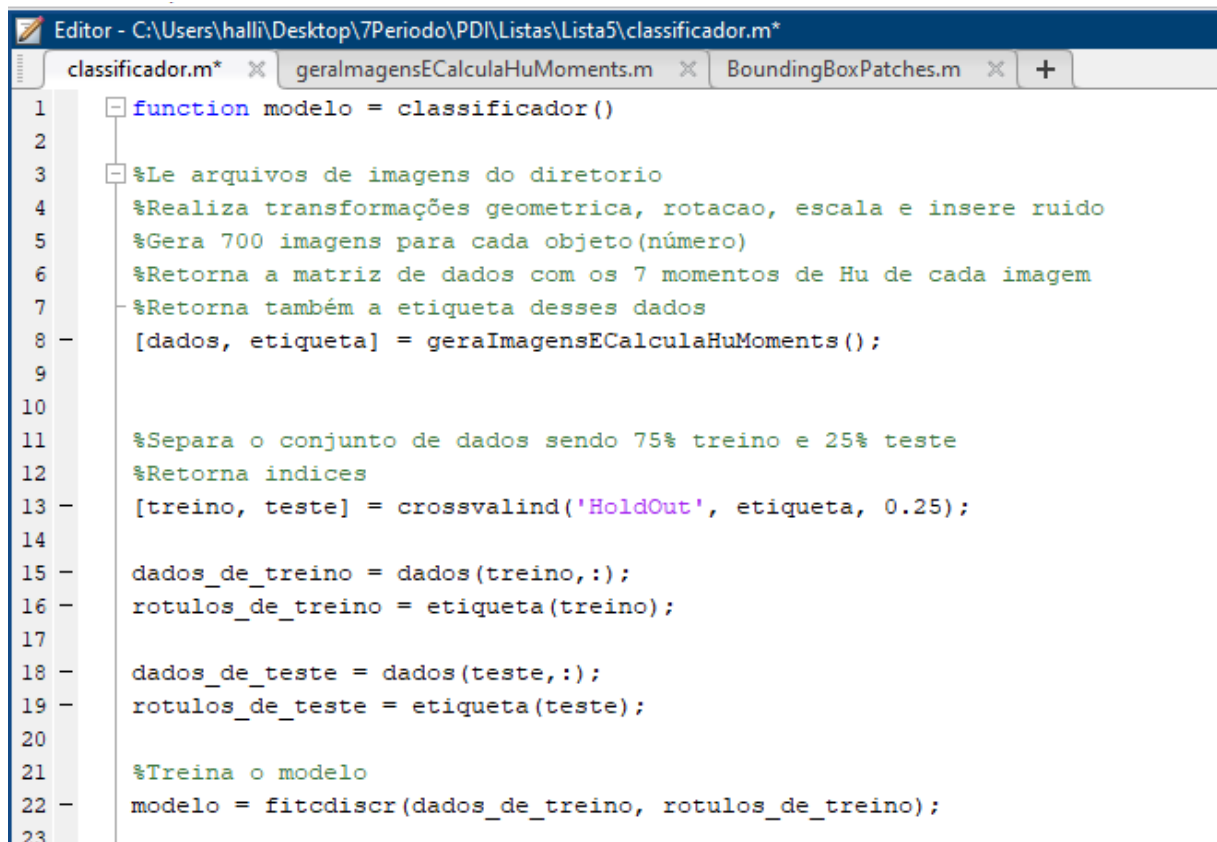
Em seguida, através do comando 'dir' é armazenado quais são o nome dos arquivos das imagens. A matriz de dados é criada e possuirá 7000 linhas e 7 colunas, cada linha armazenará os 7 momentos de Hu de uma imagem. Também é criada a matriz de etiquetas que é responsável de identificar qual número os momentos de Hu estão representando.

```
Editor - C:\Users\hall\Desktop\7Periodo\PD\Listas\Lista5\geralmagensECalculaHuMoments.m*
geralmagensECalculaHuMoments.m*  humoments.m  BoundingBoxPatches.m  amostra.m

17 - elemento = 1;
18 - %para cada imagem lida do diretorio
19 - for i = 1 : quant_img
20 -     img = imread (fullfile(endereco , arquivos(i ,1).name));
21 -     %rotacionar imagem 19 vezes de maneira aleatoria entre 1 a 359 graus
22 -     graus = 0;
23 -     for r = 1: 20
24 -         nimg = imrotate(img, graus);
25 -         graus = randi(359);
26 -         %Para cada imagem rotacionada deve-se alterar a escala
27 -         escala = 0.5;
28 -         for e = 1: 7
29 -             nimg2 = imresize(nimg, escala);
30 -             escala = escala + 0.25;
31 -             %Para cada imagem reescalada deve inserir um ruido
32 -             ruido = 0.0;
33 -             for r = 1: 5
34 -                 nimg3 = imnoise(nimg2,'salt & pepper', ruido);
35 -                 seven_moments = humoments(nimg3);
36 -                 dados(elemento,:) = seven_moments;
37 -                 etiqueta(elemento, 1) = i-1;
38 -                 elemento = elemento+1;
39 -                 ruido = ruido + 0.01;
40 -             end
41 -         end
42 -     end
43 - end
```

Seguidamente para cada imagem, a imagem é lida do diretório, rotacionada, escalada e é inserido um ruído na mesma, isso acontece de várias formas. Após essas a realização dessas transformações geométricas, é calculado os 7 momentos de Hu dessas imagens através da função `humoments`, em seguida os dados retornados são armazenados na matriz de dados e sua etiqueta é salva no vetor de etiqueta. Por fim, a função retorna a matriz de dados e o vetor de etiquetas.

Questões 5, 6 e 7)



```
Editor - C:\Users\halli\Desktop\7Periodo\PD\Listas\Lista5\classificador.m*
classificador.m* x geralimagensECalculaHuMoments.m x BoundingBoxPatches.m x +
1  function modelo = classificador()
2
3  %Le arquivos de imagens do diretorio
4  %Realiza transformações geometrica, rotacao, escala e insere ruido
5  %Gera 700 imagens para cada objeto(número)
6  %Retorna a matriz de dados com os 7 momentos de Hu de cada imagem
7  %Retorna também a etiqueta desses dados
8  [dados, etiqueta] = geralimagensECalculaHuMoments();
9
10
11  %Separa o conjunto de dados sendo 75% treino e 25% teste
12  %Retorna indices
13  [treino, teste] = crossvalind('HoldOut', etiqueta, 0.25);
14
15  dados_de_treino = dados(treino,:);
16  rotulos_de_treino = etiqueta(treino);
17
18  dados_de_teste = dados(teste,:);
19  rotulos_de_teste = etiqueta(teste);
20
21  %Treina o modelo
22  modelo = fitcdiscr(dados_de_treino, rotulos_de_treino);
23
```

A função 'classificador' vai utilizar a função geralimagensECalculaHuMoments para gerar 7000 imagens dos números de 0 a 9 (700 de cada), que sofreram transformações geométricas como rotação, escala e passaram por um processo de inserção de ruído, e gerar os momentos de Hu de cada imagem e suas respectivas etiquetas.

Em seguida a função crossvalind vai separar os dados 75% dos dados para teste e 25% dos dados para treino. Após separar os dados e suas respectivas etiquetas o modelo é treinado.

```

24 %Testa o modelo
25 - predicacao = predict(modelo, dados_de_teste);
26
27 %Conferir quanto o modelo acertou no teste
28 - [lin, ~] = size(predicacao);
29
30 - acertos = 0;
31
32 - for i = 1: lin
33 -     if predicacao(i) == rotulos_de_teste(i)
34 -         acertos = acertos + 1;
35 -     end
36 - end
37
38 %Mostra na tela
39 - disp("Porcentagem de acerto:");
40 - disp(acertos/lin);
41
42 %Gera Matriz de Confusao e mostra a mesma
43 - matrizConfusao = confusionmat(rotulos_de_teste, predicacao)
44
45 - end

```

Após treinar o modelo é feito uma predição em cima dos dados de teste e é mostrado no display a porcentagem de acerto total e a Matriz de Confusão da predição.

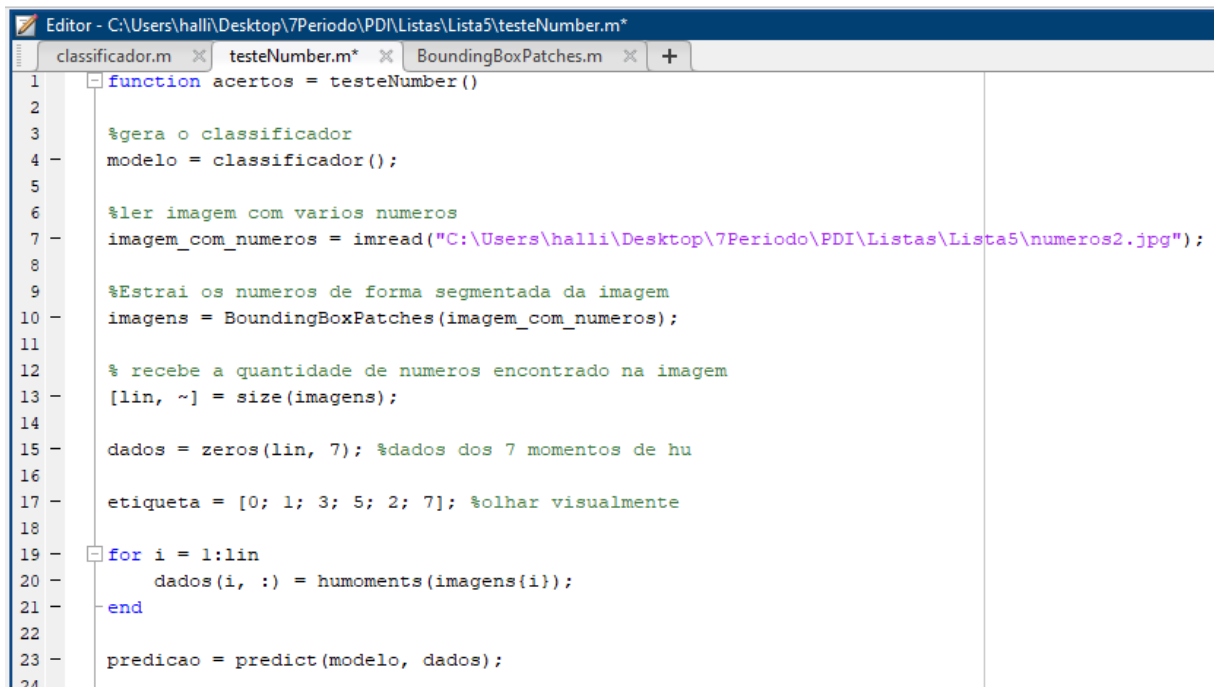
```
>> classificador
Porcentagem de acerto:
0.6594
```

```
matrizConfusao =
```

87	0	0	0	2	0	0	0	61	25
0	157	14	0	0	0	0	4	0	0
0	4	151	0	0	20	0	0	0	0
0	0	1	128	0	43	0	2	0	1
12	0	0	3	148	0	1	0	6	5
0	0	9	74	0	92	0	0	0	0
2	0	0	5	2	0	35	0	64	67
0	0	2	0	0	0	0	173	0	0
44	0	0	0	2	0	0	0	115	14
5	0	0	1	3	0	38	0	60	68

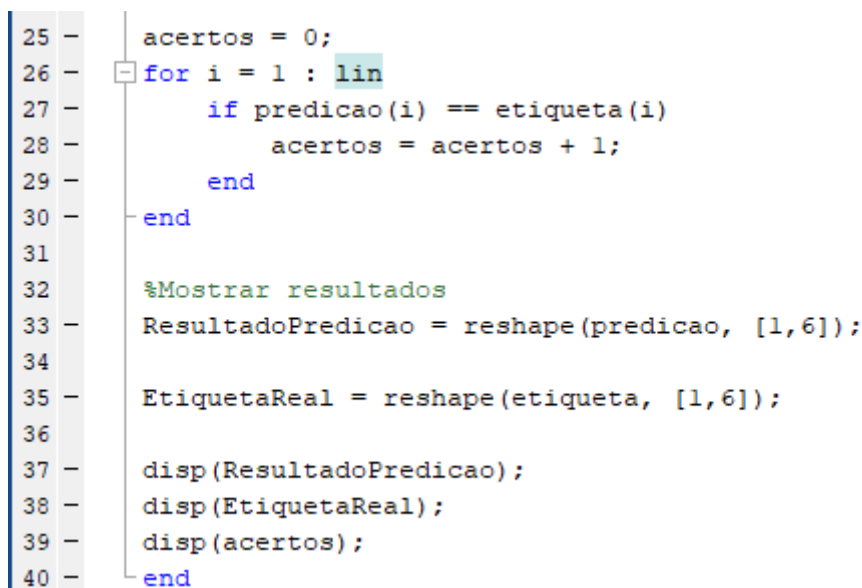
É possível observar que o classificador teve 65,94% de acerto utilizando os dados testes. Através da Matriz de Confusão é possível perceber que o modelo não acerta muito bem os números 0, 5, 6 e 9.

Questão 8)



```
Editor - C:\Users\halli\Desktop\7Periodo\PD\Listas\Lista5\testeNumber.m*
classificador.m  testeNumber.m*  BoundingBoxPatches.m  +
1  function acertos = testeNumber()
2
3      %gera o classificador
4      modelo = classificador();
5
6      %ler imagem com varios numeros
7      imagem_com_numeros = imread("C:\Users\halli\Desktop\7Periodo\PD\Listas\Lista5\numeros2.jpg");
8
9      %Estrai os numeros de forma segmentada da imagem
10     imagens = BoundingBoxPatches(imagem_com_numeros);
11
12     % recebe a quantidade de numeros encontrado na imagem
13     [lin, ~] = size(imagens);
14
15     dados = zeros(lin, 7); %dados dos 7 momentos de hu
16
17     etiqueta = [0; 1; 3; 5; 2; 7]; %olhar visualmente
18
19     for i = 1:lin
20         dados(i, :) = humoments(imagens{i});
21     end
22
23     predicacao = predict(modelo, dados);
24
```

A função testeNumber gera o modelo(classificador) através da função classificador supracitada, lê uma imagem com vários números e segmenta essa imagem através da função BoudingBoxPatches gerando várias imagens. Para cada uma dessas imagens é feito o cálculo dos seus 7 momentos de HU e é feita a predição.



```
25     acertos = 0;
26     for i = 1 : lin
27         if predicacao(i) == etiqueta(i)
28             acertos = acertos + 1;
29         end
30     end
31
32     %Mostrar resultados
33     ResultadoPredicao = reshape(predicacao, [1,6]);
34
35     EtiquetaReal = reshape(etiqueta, [1,6]);
36
37     disp(ResultadoPredicao);
38     disp(EtiquetaReal);
39     disp(acertos);
40     end
```

Após fazer a predição é mostrado no display o resultado da predição, o resultado real, e a quantidade de acertos, respectivamente.



8 1 3 5 2 7

0 1 3 5 2 7

5

É possível perceber que o classificador errou somente uma vez, classificando o número 0 como o número 8.