



Extração de Características:

Contagem de Pixels

Data de Entrega: 30/04/2023

Base de Imagens (Digitos) - [Download](#)

***Altere os arquivos.**

****Verifiquem se todas as imagens estão abrindo perfeitamente no Python.;**

Observações:

- 1 - A base são de dígitos impressos de 0 - 9 ;
- 2 - Cada pasta refere-se a um dígito (temos 10 diferentes espécies) ;
- 3 - As 100 imagens dentro de cada pasta são de amostras da mesma classe ;
- 6 - Temos 10 classes (0 - 9) ;
- 7 - Temos os arquivos de treino e teste (ambos com os mesmos números) .

O que queremos demonstrar:

- a) Amostras da mesma classe (pasta) possuem maior similaridade (menor distância euclidiana) entre elas ;
- b) Amostras entre diferentes classes (pastas) possuem maior diferença (maior distância euclidiana) entre elas ;
- c) Amostras da mesma classe (mesmo exemplo só que com alguma alteração) deve possuir vetores de características praticamente iguais ;

Crie uma conta no mathworks (Matlab) para resolver os exercícios online. É possível usar a versão online para estudante (A UTFPR comprou a licença do MATLAB, basta entrar com o e-mail da UTFPR - Pode usar a versão online!).

1 - Conte o número de pixels brancos e pretos de cada imagem ;

2 - Avalie para diferentes imagens da mesma classe o número de pixels é similar?

3 - Normalize os dados usando minmax ou z-score.

3 - E se dividirmos a imagem em quadrantes e contarmos o número de pixels pretos e brancos de cada quadrante? Utilize: 2x2, 3x3, 5x5.

5 - Gere um arquivo .txt com o número pixels pretos e brancos de cada imagem em cada linha e no final da linha coloque o nome da classe que essas features representam (0 - 9). Faça isso para cada conjunto de treinamento e teste e para cada quadrante, 1x1, 2x2, 3x3, 5x5.

Veja que na imagem 1x1, teremos 2 colunas e 1000 linhas para cada arquivo (treino / teste) ;
Na imagem 2x2, teremos 8 colunas e 1000 linhas para cada arquivo (treino / teste) ;
Na imagem 3x3, teremos 18 colunas e 1000 linhas para cada arquivo (treino / teste) ;
Na imagem 5x5, teremos 50 colunas e 1000 linhas para cada arquivo (treino / teste) ;

Você vai enviar os 8 arquivos (1x1, 2x2, 3x3, 5x5 - Treino), (1x1, 2x2, 3x3, 5x5 - Teste).

Exemplo de código Matlab

```
#####
```

```
function [] = geraBlocos(I, qtdeFragmentosVerticais, qtdeFragmentosHorizontais)
[height, width] = size( I );
height = height / qtdeFragmentosVerticais;
width = width / qtdeFragmentosHorizontais;
nomeArquivo = 'treinamento' ; % Treino e teste
arquivo = strcat(nomeArquivo, '.txt');
fid = fopen(arquivo,'w');
for idxSubImagemVertical=0:qtdeFragmentosVerticais-1
    yInicial = round( idxSubImagemVertical * height);
    yFinal = round((idxSubImagemVertical + 1) * height);
    for idxSubImagemHorizontal=0:qtdeFragmentosHorizontais-1
        xInicial = round( idxSubImagemHorizontal * width);
        xFinal = round((idxSubImagemHorizontal + 1) * width);
        subImagem = imcrop(I, [xInicial yInicial round(xFinal - xInicial) round(yFinal -
yInicial)]);
        %imshow(subImagem) ;
        % Escreva seu código aqui!!!
        % Exemplo de como imprimir no arquivo.
        fprintf(fid, '%f ', vetorFeatures );
        fprintf(fid, '%i ', classe ); % Lembre-se a classe pode ser o nome da pasta [0 -
9]
    end
end
fclose ( fid );
end
```

Segue um exemplo para varrer todas as pastas...

```
function [] = main(nomeArquivoSaida, folder, numBlocosV, numBlocosH)
dirListing = dir(folder);
arquivo = strcat(nomeArquivoSaida, '.txt');
fid = fopen(arquivo,'w');
for d = 3:length(dirListing)

    if (dirListing(d).isdir == 1)

        fileName = fullfile(folder,dirListing(d).name);
```

```

fopen(fileName);

arquivos = dir(fileName);

for i = 3 : length(arquivos)

    if (arquivos(i).isdir == 0)

        nomeArquivo = fullfile(fileName,arquivos(i).name) ;
        fopen(fileName);
        nomeClasse = fileName(end: end) ;
        %nomeArquivo
        I = imread(nomeArquivo);
        disp(nomeArquivo)
        disp(fileName)
        %[l,c] = size(I) ;

        geraBlocos(I, nomeClasse, numBlocosV,numBlocosH, fid ) ;

    end

end

end

end
fclose ( fid );

```