

Universidade Federal de Pelotas

Curso: Ciência da Computação

Disciplina: Processamento Digital de Imagens

Professor: Bruno Zatt

Relatório 7: Cálculo de Métricas de Qualidade e Entropia em Imagens

Aluno: Yago Martins Pintos

Data: 19/10/2024

1. Introdução

Este relatório descreve a implementação de duas métricas de qualidade: PSNR (Peak Signal-to-Noise Ratio) e MSE (Mean Square Error), além do cálculo da entropia de uma imagem.

2. Cálculo do MSE e PSNR

2.1 Descrição das Métricas

- MSE (Mean Square Error): É uma medida da média das diferenças quadráticas entre os valores correspondentes dos pixels de duas imagens. Quanto menor o MSE, mais semelhantes são as imagens.
- PSNR (Peak Signal-to-Noise Ratio): É uma medida que expressa a relação entre a potência do sinal (imagem original) e a potência do ruído (diferença entre as imagens). Valores mais altos de PSNR indicam melhor qualidade da imagem.

Implementação do Cálculo

```
matlab
% Leitura das imagens
imagem1 =
imread('C:\Users\yagom\projects\PDI\Tarefa7\lena_contraste.bmp'
');
imagem2 =
imread('C:\Users\yagom\projects\PDI\Tarefa7\barbara_gray.bmp')
;

% Conversão para escala de cinza
if size(imagem1, 3) == 3
    imagem1 = rgb2gray(imagem1);
end
```

```

if size(imagem2, 3) == 3
    imagem2 = rgb2gray(imagem2);
end

% Verificação da mesma resolução
if size(imagem1) ~= size(imagem2)
    error('As imagens devem ter a mesma resolução.');
```

```

end

% Conversão para double para cálculo preciso
imagem1 = double(imagem1);
imagem2 = double(imagem2);

% Cálculo do MSE (Mean Square Error)
mse = mean((imagem1(:) - imagem2(:)).^2);

% Cálculo do PSNR (Peak Signal-to-Noise Ratio)
if mse == 0
    psnr = Inf;
else
    max_pixel_value = 255; % Para imagens de 8 bits
    psnr = 10 * log10((max_pixel_value^2) / mse);
end

fprintf('MSE: %.4f\n', mse);
fprintf('PSNR: %.4f dB\n', psnr);

```

3. Cálculo da Entropia

3.1 Descrição da Entropia

A entropia é uma medida que quantifica a quantidade de informação contida em uma imagem. Uma entropia mais alta indica uma maior complexidade e variedade na imagem, enquanto uma entropia mais baixa indica áreas homogêneas.

Implementação do Cálculo da Entropia

```

matlab
function H = calcular_entropia(imagem)

    if size(imagem, 3) == 3
        imagem = rgb2gray(imagem);
    end

    histograma = imhist(imagem);

    probabilidades = histograma / numel(imagem);

    probabilidades(probabilidades == 0) = []; % Remove zeros
    para evitar log(0)

    H = -sum(probabilidades .* log2(probabilidades));

    fprintf('A entropia da imagem é: %.4f\n', H);
end

```

4. Resultados

4.1 Resultados do MSE e PSNR

Após executar o código para calcular MSE e PSNR, os resultados foram exibidos na tela:

- MSE: Valor calculado que representa a média das diferenças quadráticas entre as duas imagens.
- PSNR: Valor em decibéis que indica a qualidade da imagem comparada à imagem original.

4.2 Resultado da Entropia

A função `calcular_entropia` foi chamada com uma imagem em escala de cinza, resultando na exibição da entropia calculada:

- Entropia: Valor que quantifica a informação contida na imagem.

5. Conclusões

As métricas implementadas demonstraram ser eficazes na avaliação da qualidade das imagens:

- MSE e PSNR: Essas métricas são úteis para comparar a qualidade de diferentes versões de uma imagem, permitindo avaliar o impacto de processos como compressão ou filtragem.
- Entropia: A medida da entropia fornece insights sobre a complexidade e a quantidade de informação presente em uma imagem.

Figuras:

- Figura 1: Código para cálculo do MSE e PSNR.

```
%Utilize os passos abaixo para implementar as métricas de
qualidade PSNR
%(Peak Signal-to-Noise Ratio) e MSE (Mean Square Error)
%• Receba duas imagens em escala de cinza;
%• Garanta que tenham a mesma resolução;
%• Implemente uma função (ou duas funções separadas) que calcule
as
%equações abaixo;
% Exiba os resultados de MSE e PSNR.

imagem1 =
imread('C:\Users\yagom\projects\PDI\Tarefa7\lena_contraste.bmp');
imagem2 =
imread('C:\Users\yagom\projects\PDI\Tarefa7\barbara_gray.bmp');

if size(imagem1, 3) == 3
    imagem1 = rgb2gray(imagem1);
end

if size(imagem2, 3) == 3
    imagem2 = rgb2gray(imagem2);
end

if size(imagem1) ~= size(imagem2)
    error('As imagens devem ter a mesma resolução.');
```

```

•
•  imagem1 = double(imagem1);
•  imagem2 = double(imagem2);
•
•  % Cálculo do MSE (Mean Square Error)
•  mse = mean((imagem1(:) - imagem2(:)).^2);
•
•  % Cálculo do PSNR (Peak Signal-to-Noise Ratio)
•  if mse == 0
•      psnr = Inf;
•  else
•      max_pixel_value = 255; % Para imagens de 8 bits
•      psnr = 10 * log10((max_pixel_value^2) / mse);
•  end
•
•  fprintf('MSE: %.4f\n', mse);
•  fprintf('PSNR: %.4f dB\n', psnr);
•
•

```

• **Figura 2: Código para cálculo da entropia.**

```

•  %Utilize os passos abaixo para implementar o cálculo da entropia
•  de uma
•  %imagem
•  % Receba uma imagem em escala de cinza;
•  % Implemente uma função que calcule as equações abaixo;
•  %OBS: Log é logaritmo de base 2
•  % Exiba os resultados da entropia H.
•  function H = calcular_entropia(imagem)
•
•
•      if size(imagem, 3) == 3
•          imagem = rgb2gray(imagem);
•      end
•
•
•
•
•      histograma = imhist(imagem);
•
•

```

```

•
•   probabilidades = histograma / numel(imagem);
•
•
•   probabilidades(probabilidades == 0) = [];
•
•
•   H = -sum(probabilidades .* log2(probabilidades));
•
•
•   fprintf('A entropia da imagem é: %.4f\n', H);
•   end

```

-
-
- Figura 3: Resultados exibidos para MSE, PSNR e Entropia após execução dos códigos.

MSE: 4901.8056

PSNR: 11.2272 dB

>> calcular_entropia('barbara_gray.bmp')

A entropia da imagem é: -0.0000

ans = 0