

Relatório - Trabalho 1

Júlia Yuri Garcia Baba
Departamento de Ciência da
Computação
UnB
Brasília, Brasil
190057921@aluno.unb.br

Keywords – interpolação, filtro, equalização

I RESUMO

Este trabalho teve como objetivo principal aplicar os conhecimentos obtidos em sala de aula sobre processamento e características gerais de imagens, como redimensionamento, distribuição de brilho, cor e contraste. Os códigos foram todos desenvolvidos em OpenCv 4.5.1, usando a linguagem de programação Python, sendo testados com imagens distintas.

A primeira parte do projeto consiste em alterar a imagem para uma versão menor dela e posteriormente interpolar para o tamanho original, repetindo o pixel mais próximo. Assim como, aplicar um filtro de aguçamento, afim de melhorar subjetivamente a qualidade da imagem e propor uma maneira mais efetiva de realizar a interpolação. Já a segunda parte, tem o objetivo de testar a capacidade de dois métodos, correção gamma e equalização, de redistribuir brilho mais efetivamente.

As principais dificuldades enfrentadas durante o desenvolvimento foram a implementação sem o uso das funções pré-existentes, tal qual como o uso da linguagem

II INTRODUÇÃO

A.Primeira Questão: Redimensionamento de imagens e aplicação de um filtro de aguçamento espacial

A interpolação pelo vizinho mais próximo é a forma mais simples de interpolação. Este método consiste simplesmente em determinar o pixel vizinho mais próximo e assumir o seu valor de intensidade. Dada a imagem, em sua versão colorida (RGB), usaremos esse método para criar a função que irá redimensionar a figura, tanto para um tamanho menor (fator N, múltiplo de 2), como para a volta à seu tamanho original.

Logo, para a segunda parte da questão, foi escolhido um filtro baseado na derivada de segunda ordem: o operador laplaciano. Ao contrário dos filtros de primeira ordem, que detectam as arestas com base em máximos e mínimos locais, o Laplaciano detecta as arestas em cruzamento de zeros, ou seja, aonde o valor muda de positivo para negativo, e vice-versa. Assim, realça as descontinuidades entre níveis de cinza e tira a ênfase de transições suaves.

Note que o programa principal deve ter como saída duas interpolações, vizinho mais próximo e bicúbica, assim como a aplicação do filtro de aguçamento espacial para ambas imagens geradas, que serão apresentadas e analisadas na seção de Resultados.

B.Segunda Questão: Correção e equalização de imagens

Já na segunda parte trabalho, o propósito consiste em corrigir imagens monocromáticas através de métodos de ajuste de brilho e dessa forma, verificar a suas respectivas eficácias.

Tanto a correção gamma, quanto a equalização, tem o papel de ajustar o contraste da figura, consertando possíveis falhas em relação ao branqueamento ou escurecimento. Contudo, a equalização é um procedimento no qual essa regulagem é feita através do histograma da imagem, enquanto a correção gamma é dada matematicamente por:

$$s = c \cdot r^\gamma$$

Onde s e r são respectivamente os valores de pixel de saída e entrada, e c e γ são constantes positivas

III METODOLOGIA

Dado o resumo dos principais pontos de cada exercício, será feita uma explicação breve de como cada objetivo foi alcançado. Quais funções foram usadas e quais algoritmos foram implementados.

A. Primeira Questão:

a) Redimensionamento

O redimensionamento da imagem foi feito através de um algoritmo simples, onde linhas e colunas da imagem são removidas ou adicionadas em um intervalo proporcional à proporção da mudança das dimensões do tamanho da imagem, ou seja, quando é necessário adicionar uma linha ou coluna, essas são copiadas da linha ou coluna anterior a posição que se deve ser adicionado.

Para efetuar tal ação, primeiramente, lemos e armazenamos os valores de altura e largura da imagem em variáveis que serão usadas para calcular as novas dimensões da figura, essas serão encontradas através da divisão por um fator N, múltiplo de dois. Depois de efetuadas as operações, com as novas dimensões, calculamos os fatores de redimensionamento e usando a biblioteca numpy, criamos uma matriz de zeros com as novas proporções.

Para obtermos a imagem dizimada, utilizamos um algoritmo que percorre, pixel por pixel, da matriz de zeros e a preenche com o valor do pixel vizinho mais próximo da figura original. Onde a linha e coluna do pixel é acessada por linha e coluna da saída, dividido pelos fatores de redimensionamento e arredondados com a função *floor* da biblioteca math.

Já para a obtenção da imagem interpolada fazemos o mesmo processo, porém com pequenas diferenças. Como

queremos voltar a imagem para o seu tamanho original, devemos usar a largura e altura da imagem dizimada e calcular as novas dimensões, que serão iguais a da figura original, contudo faremos a operação inversa, ao invés de dividir, multiplicaremos por um fator N. Dessa forma, podemos obter os novos fatores de redimensionamento e interpolar a imagem para o tamanho inicial.

Note que, quando diminuimos a imagem, estamos na prática perdendo uma certa quantidade de informações e ao interpolarmos essa mesma imagem para o seu tamanho original estamos reproduzindo o pixel mais de uma vez. Contudo, os dados perdidos não são repostos, mas sim, substituídos por cópias dos pixels da imagem dizimada.

A interpolação bicúbica, que foi chamada através de uma função pré-existente do OpenCv, considera 16 pixels (4 x 4) ao invés de 4 pixels (2 x 2). Desse modo, afeta diretamente a volta da imagem para o tamanho original, já que menos dados serão perdidos, ou seja, haverá mais pixels para fazer as devidas substituições.

b) Filtro de aguçamento

Após feitas tais interpolações, o filtro de aguçamento busca melhorar a qualidade subjetiva das imagens. Da explicação introdutória, foi dito que a segunda derivada pode ser usada para detectar arestas. Como as imagens são “2D”, precisaríamos obter a derivada em ambas as dimensões. Para isso usamos o operador Laplaciano:

$$Laplace(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Uma vez que a imagem de entrada é representada como um conjunto de pixels discretos, temos que encontrar um kernel de convolução discreto que possa aproximar as segundas derivadas na definição do Laplaciano. Primeiramente, obtemos dois elementos de kernel, usando o mesmo processo de diferenciação das derivadas de primeira ordem, adicionando os resultados obtidos temos o chamado kernel laplaciano negativo, pois seu pico central é menor que zero.

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & -2 & 1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Observe que, o Laplaciano deve ser aplicado a uma imagem que foi primeiro convertida em nível de cinza e suavizada, por um filtro de suavização gaussiano, para reduzir sua sensibilidade ao ruído, o que foi feito, através de duas funções pré-existentes do OpenCv: *cv2.cvtColor()* e *cv2.GaussianBlur()*. Logo após esse procedimento, foi aplicado o filtro de aguçamento laplaciano, também por meio de um função do OpenCv: *cv2.Laplacian()* e feito o cruzamento de zeros, cortando os valores negativos, já que desse modo podemos localizar as bordas pelo lado com mais contraste

B. Segunda Questão:

a) Correção gama

A correção gama é transformação de intensidade, não linear, que análoga a fórmula previamente citada, é dada por:

$$I' = 255 \times (I / 255)^\gamma$$

Poderemos notar, analisando matematicamente, que gamma e intensidade possuem uma relação inversamente

proporcional. Dessa forma, quando $\gamma > 1$, a intensidade dos pixels diminui, ou seja, a imagem ficará mais escura. Por outro lado, $\gamma < 1$, a intensidade aumenta, ou seja, a imagem ficará mais clara. Potanto, foram testados para cada imagem quatro valores de gama, dois menores, e dois maiores que um. As saídas consistem no melhor resultado de cada teste.

b) Equalização do histograma

Esse método consiste geralmente em aumentar o contraste global das imagens, estendendo a faixa de intensidade da imagem, distribuindo efetivamente os valores de intensidade mais frequentes. Dessa forma, por meio desses ajuste as intensidades podem ser melhor distribuídas no histograma, permitindo que áreas de menor contraste local ganhem um contraste mais alto, de forma mais uniforme.

Assim, a saída desejada é o histograma e o CDF (função de distribuição acumulada) de uma das imagens, antes e depois da aplicação da equalização, esse resultado pode ser obtido através do uso de funções da biblioteca matplotlib.

IV RESULTADOS

Após uma noção geral de cada objetivo dos exercícios propostos e de como cada questão foi implementada, será apresentada nessa seção a saída de cada um dos programas com imagens pré-definidas

A. Primeira Questão:



Fig.1. Resultado do redimensionamento da imagem. (a) Imagem reduzida a metade do seu tamanho original utilizando o método de interpolação por vizinho mais próximo. (b) Imagem reduzida a metade do seu tamanho original utilizando o método de interpolação bicúbica.

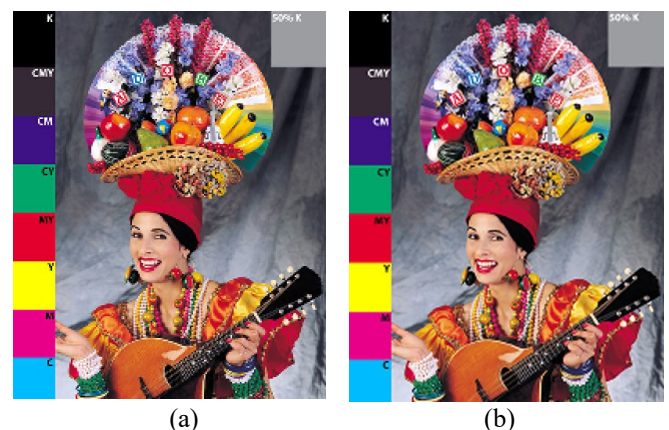


Fig.2. Resultado da interpolação da imagem dizimada. (a) Imagem de volta ao seu tamanho original utilizando o método de interpolação por vizinho mais próximo. (b) Imagem de volta ao seu tamanho original utilizando o método de interpolação bicúbica

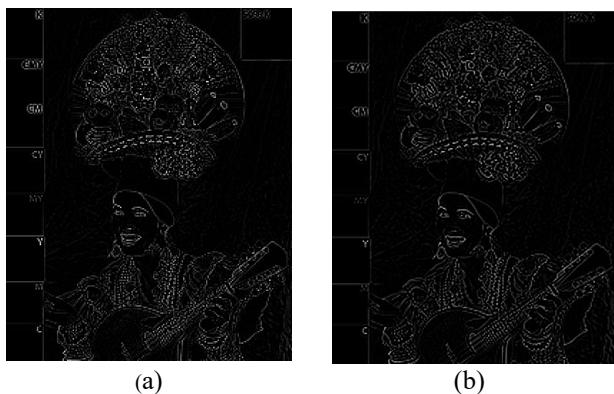


Fig.3. Resultado da aplicação do filtro de aguçamento. (a) Imagem após aplicação do filtro, antes interpolada por método do vizinho mais próximo (b) Imagem após aplicação do filtro, antes interpolada bicubicamente

Nota-se uma perda de qualidade em ambas as imagens, contudo, na interpolação por vizinho mais próximo se destaca um agravamento maior nas bordas.

B. Segunda Questão

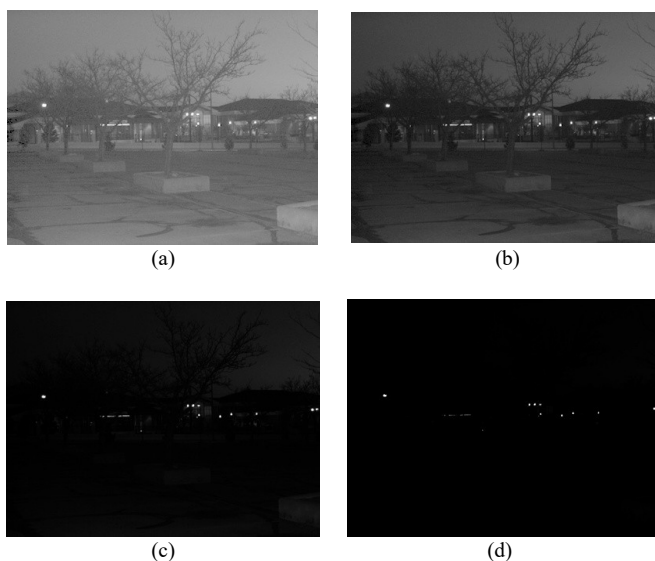
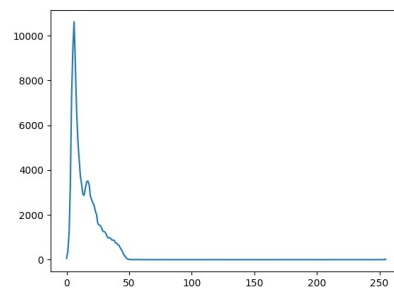


Fig.4. Resultado após a correção gamma. (a) Imagem com fator de correção gamma = 0.25 (b) Imagem com fator de correção gamma = 0.5. (c) Imagem com fator de correção gamma = 1.2. (d) Imagem com fator de correção gamma = 2.4

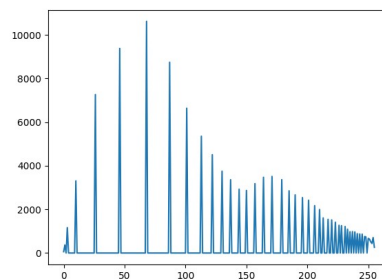
Demonstra-se através dos resultados dos teste que dependendo do valor de gama a imagem fica mais claro ou escura



Fig.5. Resultado após a equalização. (a) Imagem antes da equalização (c) Imagem depois da equalização

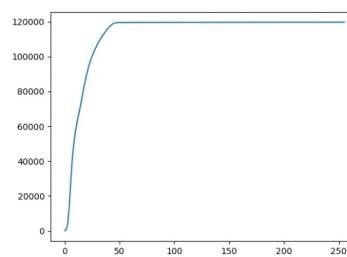


(a)

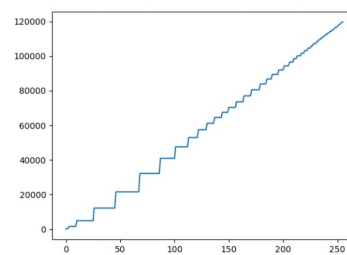


(b)

Fig.6. Histogramas. (a) Histograma antes da equalização. (b) Histograma após a equalização



(a)



(b)

Fig.7. CDF. (a) CDF antes da equalização. (b) CDF após a equalização

Após realizar a equalização do histograma, é possível notar na figura 5.(b) que a imagem está mais nítida. Analisando seu histograma e CDF, conseguimos identificar claramente que os pixels se espalharam pelos níveis de cinza, permitindo assim, esta melhora.

V CONCLUSÃO

A partir dos resultados obtidos, podemos notar a perda na qualidade das imagens após ambas as interpolações. Contudo, a interpolação por vizinho mais próximo é a que mais sofreu modificações, já que a funcionalidade do algoritmo causa mais perda de informações. Também pode-se observar, uma perda de

nitidez, ainda que menor, na interpolação bicúbica, levando a uma imagem desfocada.

Isso fica ainda mais claro quando aplicamos o filtro de aguçamento espacial nas imagens, já que as linhas ressaltadas são, em geral, mais suaves na figura 3.(b), fazendo com que mais detalhes sejam mantidos.

Já na segunda parte do experimento, vemos claramente a relação do valor de gamma com a intensidade e como ao aplicar a correção gama podemos aprimorar o brilho e o contraste, tornando as imagens mais brilhantes e com uma aparência mais natural, dependendo do valor de gamma escolhido.

Por fim, com a aplicação do processo de equalização de histograma é possível perceber nas imagens uma melhora significativa, especialmente no contraste. Contudo, se aplicarmos este processo em uma imagem cujos pixels já estão espalhados por toda a escala de intensidade, o efeito praticamente não será notado. Observe ainda que, apesar dos histogramas e CDF serem diferentes, as imagens são as mesmas, a única diferença está no contraste, não havendo qualquer modificação no conteúdo.