Centro Universitário Ritter dos Reis

Gabriela Von

João Vitor Madrid Beck

Maurício Carvalho dos Santos

**Relatório**

Porto Alegre 2023

**Etapa I**

O artigo escolhido trata sobre o reconhecimento de soja por meio de processamento de imagens aliado a redes neurais. As imagens passam por alguns processos tanto para ficarem prontas para o processamento, quanto para obtenção do resultado final.

**Primeira Etapa:**

A primeira etapa é constituída pela aquisição de imagens para o processamento. A aquisição é feita por meio pelos cientistas com a utilização de uma câmera fotográfica modelo Coolpix995 da marca Nikon com resolução de 3.34 megapixels.

**Segunda Etapa:**

A segunda etapa é constituída da aplicação de um filtro anti-aliasing que transforma a imagem em tons acinzentados. Esta etapa é importante para a remoção de ruídos da imagem que possam vir a atrapalhar o resultado final do processamento.

**Terceira Etapa:**

A terceira etapa, é constituída pela segmentação da imagem, ou seja a aplicação de mais filtros. Por meio da detecção das bordas da semente ( Método de Prewitt ), dilatação das bordas e remoção de segmentos não necessários, é possível obter uma imagem “limpa” para o processamento.

**Quarta Etapa:**

Na quarta etapa, as imagens são representadas de maneira matricial, com o tamanho de 130x130. Isto permite que as redes neurais possam realizar o processamento. Nesta etapa, as imagens já se encontram prontas para realização da análise.

**Quinta Etapa:**

A quinta etapa se dá pela análise das redes neurais, onde a imagem passa por uma rede neural que utiliza a técnica feedforward multicamadas, com três camadas ocultas. A validação do RNA confirma que o processamento de imagens aliado as redes neurais, são capazes de identificar a variedade da soja por meio de imagens.

**Etapa II**

**Carregamento de imagem**

Para realização do carregamento da imagem, foi utilizada uma variável para armazenar o caminho absoluto de uma imagem. Logo em seguida, uma variável do tipo matriz foi utilizada e utilizada uma função com o caminho como parâmetro, para realizar a sua atribuição de valor.



*Imagem ilustrando o processo de carregamento da imagem.*

**Exibindo a imagem**

Para realizar a exibição da imagem carregada, é necessário a utilização da função *imshow* passando a imagem em forma de matriz e o nome que será exibido na janela como parâmetros.

A função *waitkey* é utilizada para manter o loop operando e não deixar a exibição da imagem encerrar. Quando o usuário pressiona alguma tecla, o programa encerra a exibição da imagem.



*Imagem ilustrando o processo de exibição da imagem.*



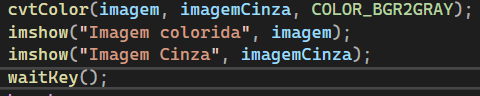
*Resultado do processo.*

**Aplicação de escala de cinza**

Para a realização da aplicação da escala de cinza, foi necessária a criação de uma outra variável do tipo matriz para armazenar a nova imagem gerada. O processo de transformação é realizada por meio de uma função ( *cvtColor* ) da biblioteca *Opencv*.

A função é responsável por utilizar a imagem original para gerar uma nova com base em um parâmetro que defina a escala de cores. Dentro do quesito solicitado, foi utilizado o parâmetro ( *COLOR\_BGR2GRAY* ) para criar a nova imagem aplicando a escala de cinza.

O código realiza a exibição da imagem original e a imagem com escala de cinza aplicada, de maneira simultânea para o usuário poder realizar a comparação.



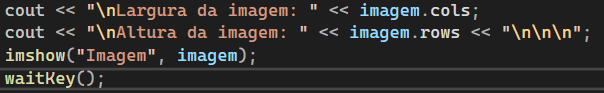
*Imagem ilustrando o processo de escala de cinza.*



*Resultado do processo.*

**Apresentação das propriedades da imagem**

O código possuí a imagem de maneira matricial, com o isso por meio de métodos da biblioteca *Opencv*, podemos realizar a exibição de informações como número de linhas e colunas da imagem.



*Imagem ilustrando a exibição de informações da imagem por meio de métodos da biblioteca Opencv.*

O método *cols* é utilizado para retornar o número de colunas da matriz da imagem, enquanto o método *rows* é utilizado para retornar o número de linhas.

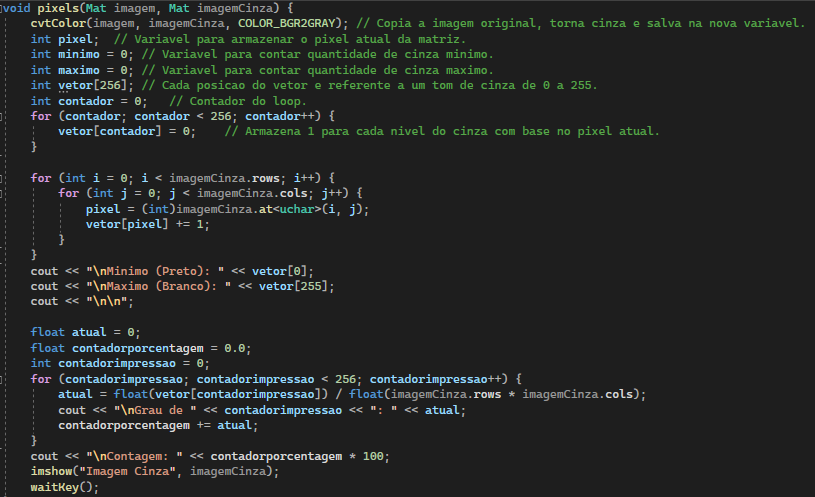


*Imagem ilustrando o resultado obtido no console.*

**Manipulando a imagem em cinza em forma de matriz**

Para realizar a verificação dos valores da imagem manipulada em cinza, criamos um método responsável por analisar pixel a pixel a matriz da imagem. Com a análise de cada pixel, seu valor de cinza é armazenado em um vetor para posteriormente ser utilizado no cálculo de porcentagem.

O método também é responsável por realizar a contagem dos valores de cinza para verificar se a imagem foi analisada de maneira correta.

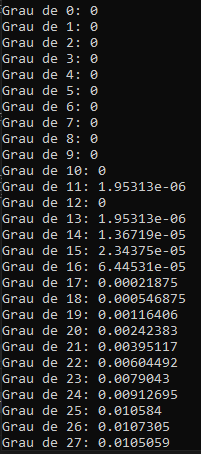


*Imagem ilustrando o método utilizado para analisar a imagem manipulada em cinza.*

O primeiro passo para realizar o processo, é a utilização da função para conversão da imagem para cinza. Após isso, o primeiro laço de repetição é responsável por armazenar zerar o valor de todos os elementos do vetor. Já o segundo laço de repetição é responsável por realizar a análise de pixel por pixel e armazenar o valor de cinza na posição do vetor de seu respectivo número. Por exemplo, se o pixel analisado possuí o valor de cinza 120, o código incrementa 1 na posição 120 do vetor de análise.

As linhas subsequentes são responsáveis por imprimir a quantidade de pixels com valor mínimo ( 0 ) e com o valor máximo ( 255 ) de cinza na imagem.

Nas linhas da parte inferior, há um laço de repetição para exibir a porcentagem de cada tonalidade de cinza presente da imagem. Ao final é realizado um cálculo para checar se a contagem foi feita corretamente.



*Imagem ilustrando a contagem de graus de cinza da imagem ( resultados em porcentagem ).*



*Imagem ilustrando o resultado final da contabilidade ( em porcentagem ).*

**Comparação dos valores e Histogramas**

**Etapa III**

**Filtros**

**Blur:**

O filtro *Blur* é uma técnica utilizada no processamento de imagem para reduzir o ruído da imagem. Para seu funcionamento, a imagem é utilizada como matriz e o efeito é aplicado *pixel* a *pixel* com base na média ponderada dos *pixels* “vizinhos”. A quantidade de *pixels* “vizinhos” a ser utilizada, é chamada de *Kernel* e pode ser alterada.

**Filtro de convolução:**

O filtro de convolução é uma técnica utilizada para preparar a imagem para um processamento. Para isso, o filtro utiliza uma imagem na forma de matriz, o kernel e também o tipo de profundidade.

O cálculo para realizar o processo, é feito pixel a pixel com uma multiplicação ponto a ponto entre os valores estabelecidos pelo kernel. O resultado das operações, é somado para obter o valor resultante no pixel da imagem de saída.

**Erosão:**

O filtro de erosão, é baseado na operação morfológica. Para seu funcionamento, cada pixel é comparado aos seus “vizinhos” e caso possuam o mesmo valor, o pixel original se mantém inalterado, caso for diferente, o pixel de saída é alterado. Um fato importante, é que o filtro de erosão pode causar a diminuição no tamanho dos objetos e também perda de detalhes na imagem caso usado de maneira incorreta.

**Comparação dos Filtros**

**Filtros no artigo**

No artigo, foram aplicados filtros para realizar a preparação para o processamento de imagem. Podemos destacar a utilização da escala de cinza para a remoção de ruídos e melhora na precisão do resultado final e também a utilização do filtro de dilatação de bordas, para auxiliar no processo de detecção dos grãos de soja.