

CENTRO UNIVERSITÁRIO RITTER DOS REIS

TRABALHO GRAU A

PROCESSAMENTO DE IMAGENS

Thomas Soares

Eduardo Amroginski

Rafael Oliveira

Leonardo Boero

1. Especificação

O trabalho consistiu no desenvolvimento de uma aplicação para experimentar operações de amostragem e quantização.

A aplicação tinha como objetivo ter a possibilidade reduzir a imagem com 3 opções de tamanhos, também tinha que ser possível definir e aplicar uma matriz, também com 3 opções, para estabelecer um valor padrão de amplitude.

Esses dois efeitos podem ser aplicados quantas vezes o usuário quiser e na ordem que for determinada por ele. Para cada efeito é possível visualizar um histograma e gerar uma imagem.

Para essa aplicação foi desenvolvida na linguagem **Python** e foi utilizadas bibliotecas como:

- **NumPy**

O NumPy é utilizado basicamente para facilitar a manipulação de vetores e matrizes.

- **OpenCV Python**

O OpenCV é uma biblioteca utilizada para carregar imagens e fazer manipulações nelas.

- **Matplotlib**

O Matplotlib tem diversas funções, no entanto a função dela no nosso experimento foi para gerar os histogramas.

Também foi utilizado um programa para auxiliar na conversão das imagens para pgm: **ImageMagick**.

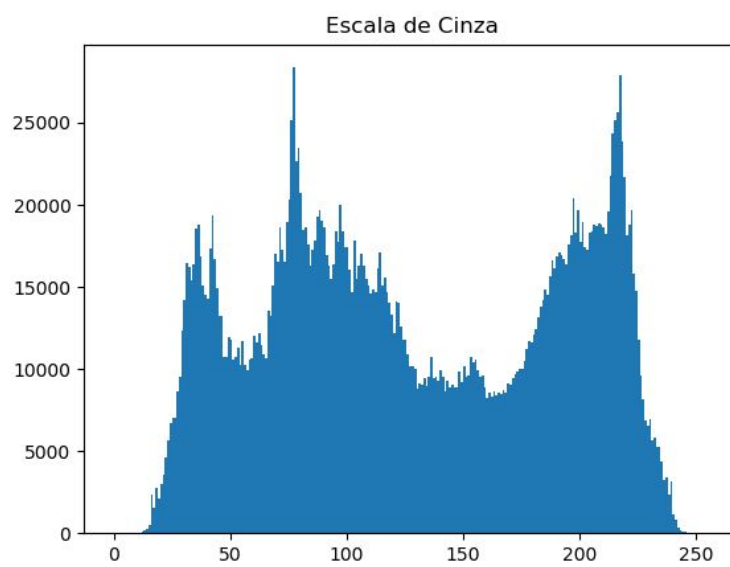
A imagem escolhida para os experimento foi essa:

Imagem Carregada (Padrão):



Imagem 1000 x 1000

E esse é o histograma da imagem:



2. Amplitude

Abaixo temos exemplos de imagens e histogramas com escalas de cinza gerados pela nossa aplicação:

2.1. Escala de imagem

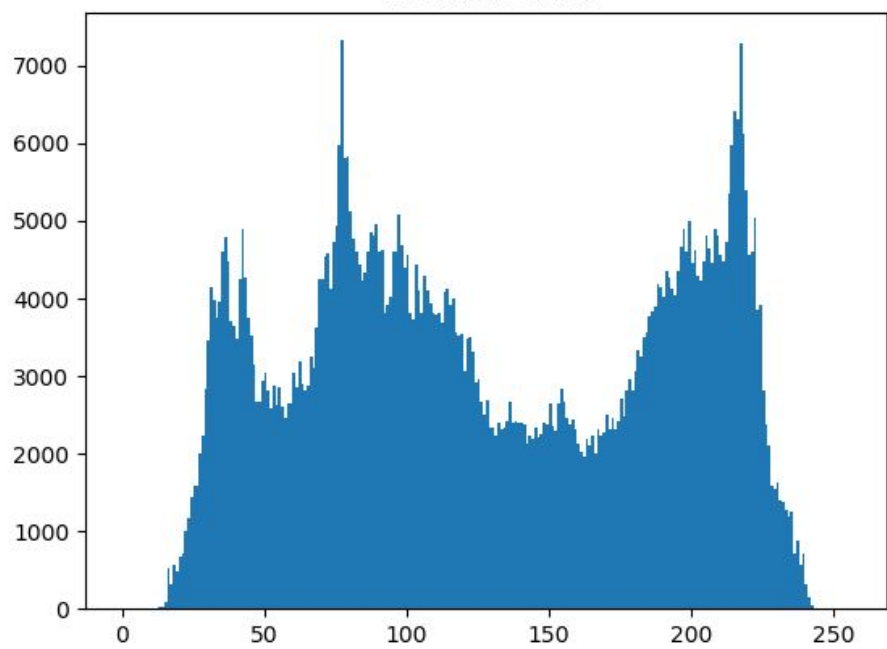
Neste exemplo o usuário selecionou a opção 2, então foi solicitado ao usuário que digite um valor menor do que 100, esse valor definirá o tamanho da nova imagem, nesse caso foi digitado 50. A função pega altura e largura da imagem e multiplica pelo valor em porcentagem que foi digitado pelo usuário.

Não houve perda de qualidade ou distorções perceptíveis na imagem, apenas sua escala foi reduzida, mantendo grande parte de suas características originais.

Teste 1– Escala de Imagem 50%



Escala de Cinza



2.2. Redimensionamento de largura

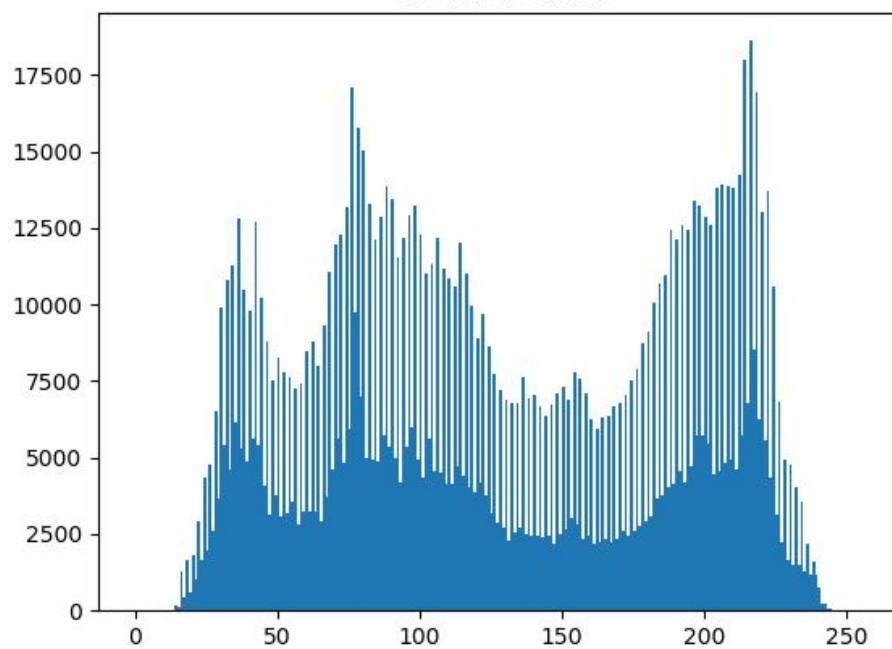
Neste exemplo o usuário selecionou a opção 3, então foi solicitado ao usuário que digite um valor menor do que a largura original da imagem, esse valor definirá o tamanho da nova imagem, nesse caso foi digitado 500. A função mantém o valor da altura e substitui o valor da largura pelo valor que foi digitado pelo usuário.

Como resultado, a imagem ficou pouco distorcida, porém ainda é possível identificar seus detalhes, e distinguir os elementos que compõem a imagem.

Teste 2 – Largura diminuída para 500px



Escala de Cinza

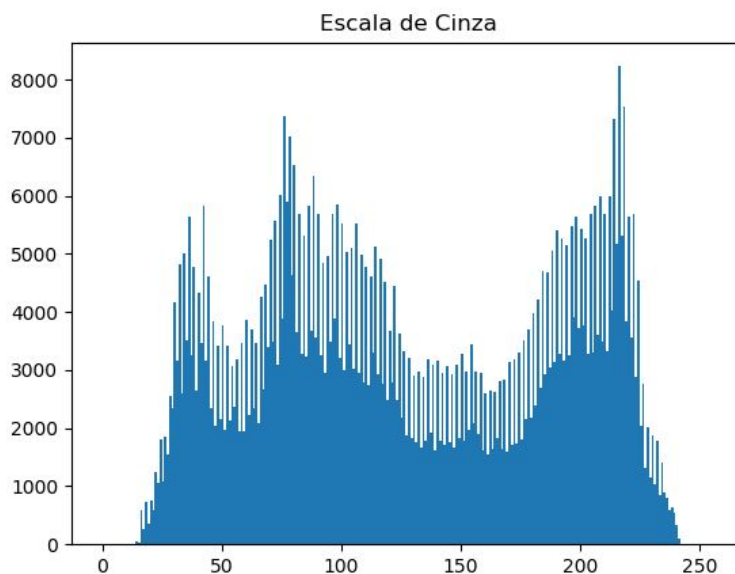
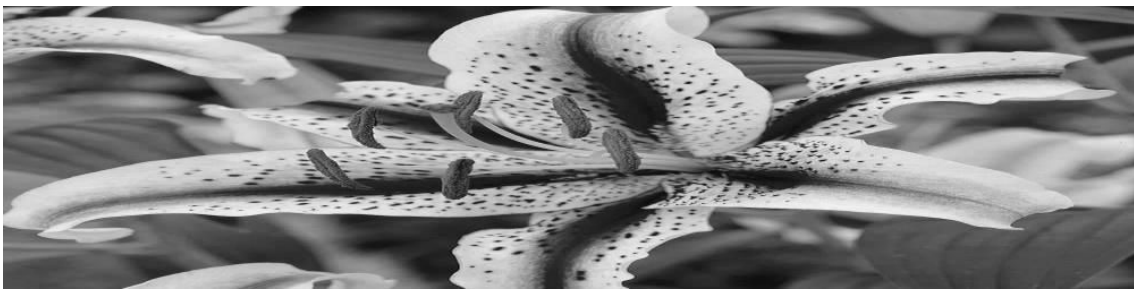


2.3. Redimensionamento de altura

Neste exemplo o usuário selecionou a opção 4, então foi solicitado ao usuário que digite um valor menor do que a altura original da imagem, esse valor definirá o tamanho da nova imagem, nesse caso foi digitado 250. A função mantém o valor da largura e substitui o valor da altura pelo valor que foi digitado pelo usuário.

Como resultado, obtemos uma imagem com bastante distorção, um certo achatamento da imagem.

Teste 3 - Altura diminuída para 250px

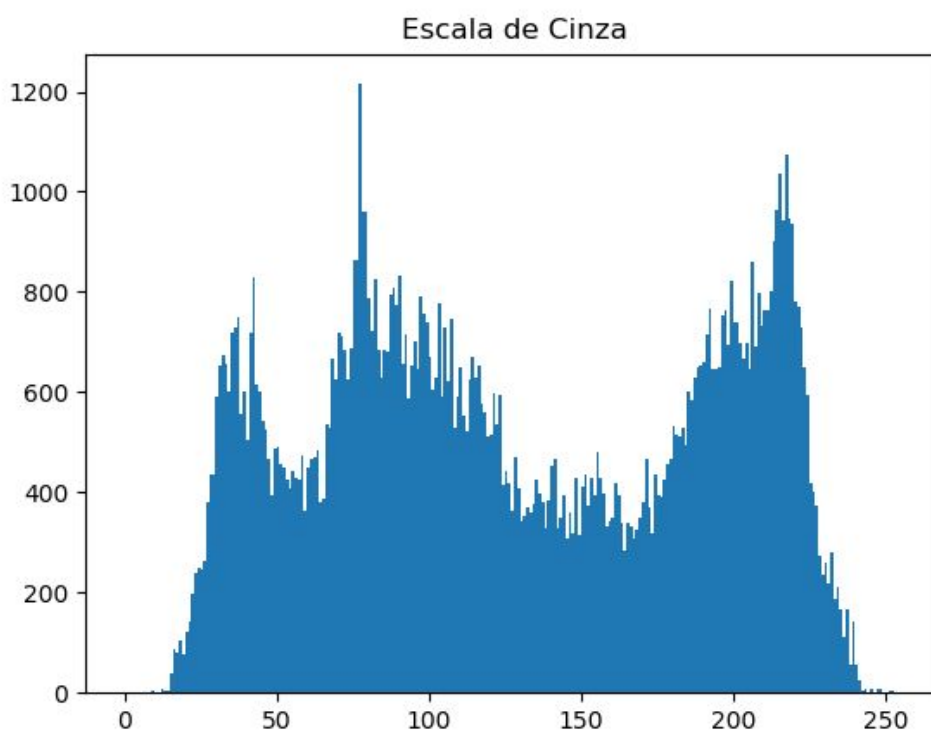


2.4. Seleção por linhas e colunas

Seleciona uma em cada N colunas, e de cada coluna uma a cada N linhas. Esse valor N é escolhido pelo usuário, e então é atribuído a seleção de linhas e colunas na matriz da imagem.

Nesse teste, utilizamos $N = 5$. Há pouca perda de qualidade, e pouca distorção na imagem.

Teste 4 - Redução linha X coluna



3. Quantização

3.1. Redução de tons de cinza - 8 tons

Neste exemplo o usuário selecionou a opção 6, que tem como função criar uma nova imagem com apenas 8 tons de cinza.

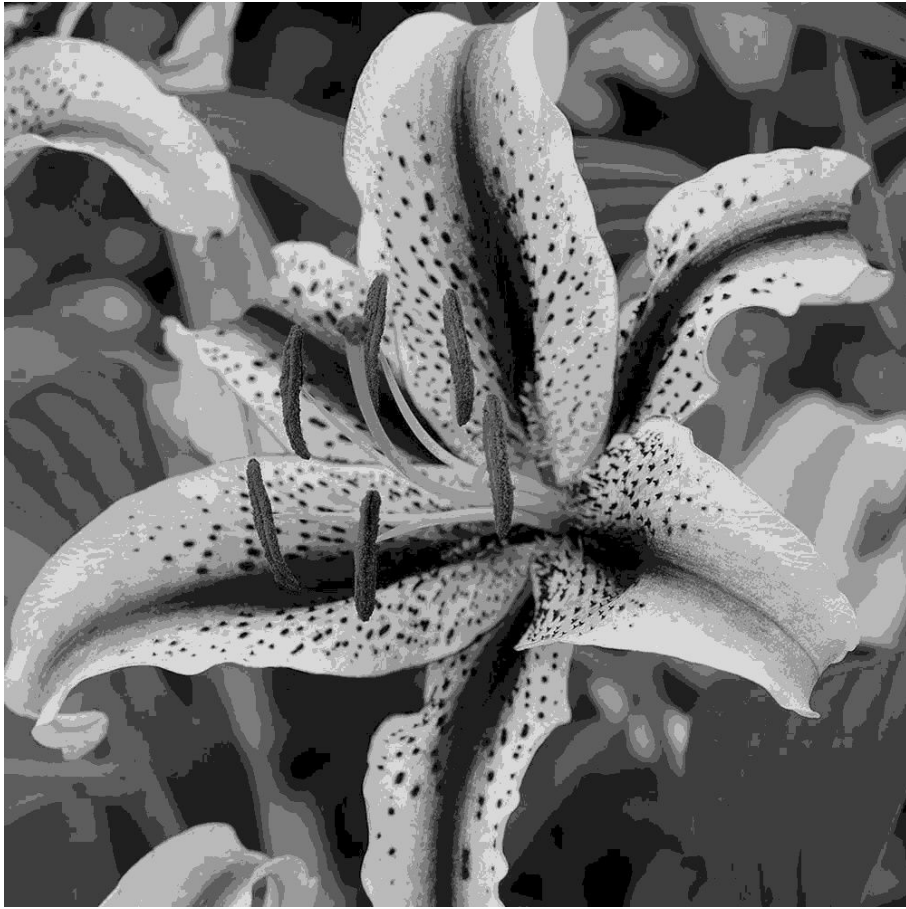
$$255 / 31 = 8,22...$$

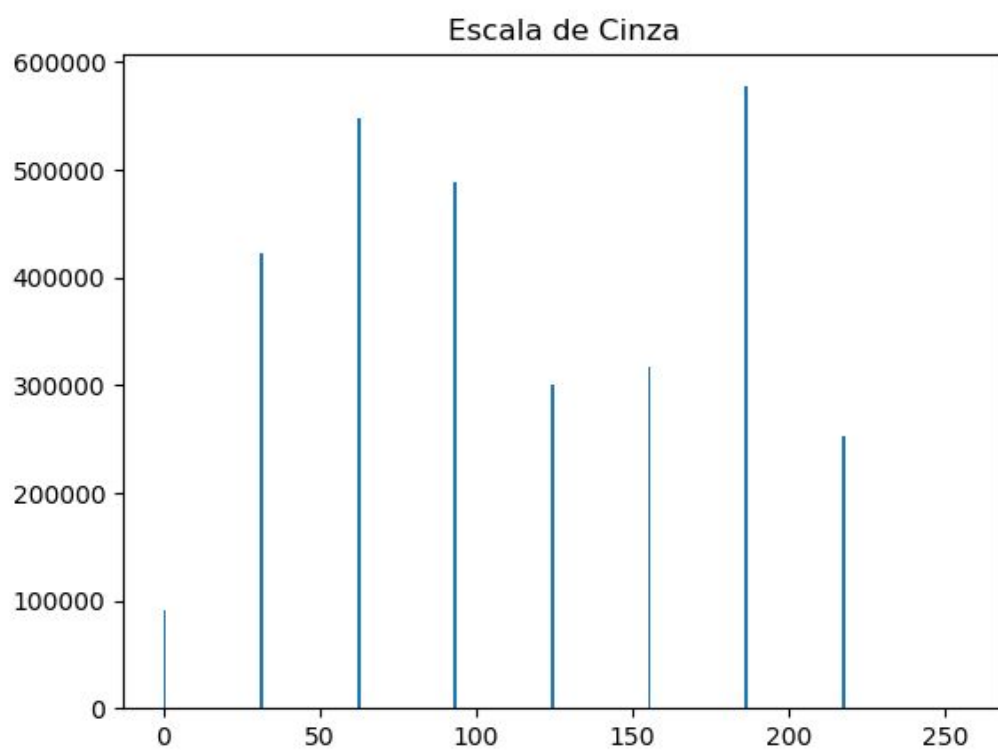
É utilizada uma função do NumPy para descartar os valores decimais.

Assim, teremos uma imagem com 8 tons de cinza. A conta é feita desta forma para descartar a parte decimal dos números e alterar o vetor para que possua apenas 8 valores possíveis.

Como resultado é aplicado uma média de cada um dos 8 tons de cinza que preenchem os demais tons removidos. Com isso, perdemos detalhes na imagem, porém ainda é possível distinguir o que é a flor e o que é o fundo.

Teste - 5 - 8 tons





3.2. Redução de tons de cinza - 6 tons

Neste exemplo o usuário selecionou a opção 6, que tem como função criar uma nova imagem com apenas 6 tons de cinza.

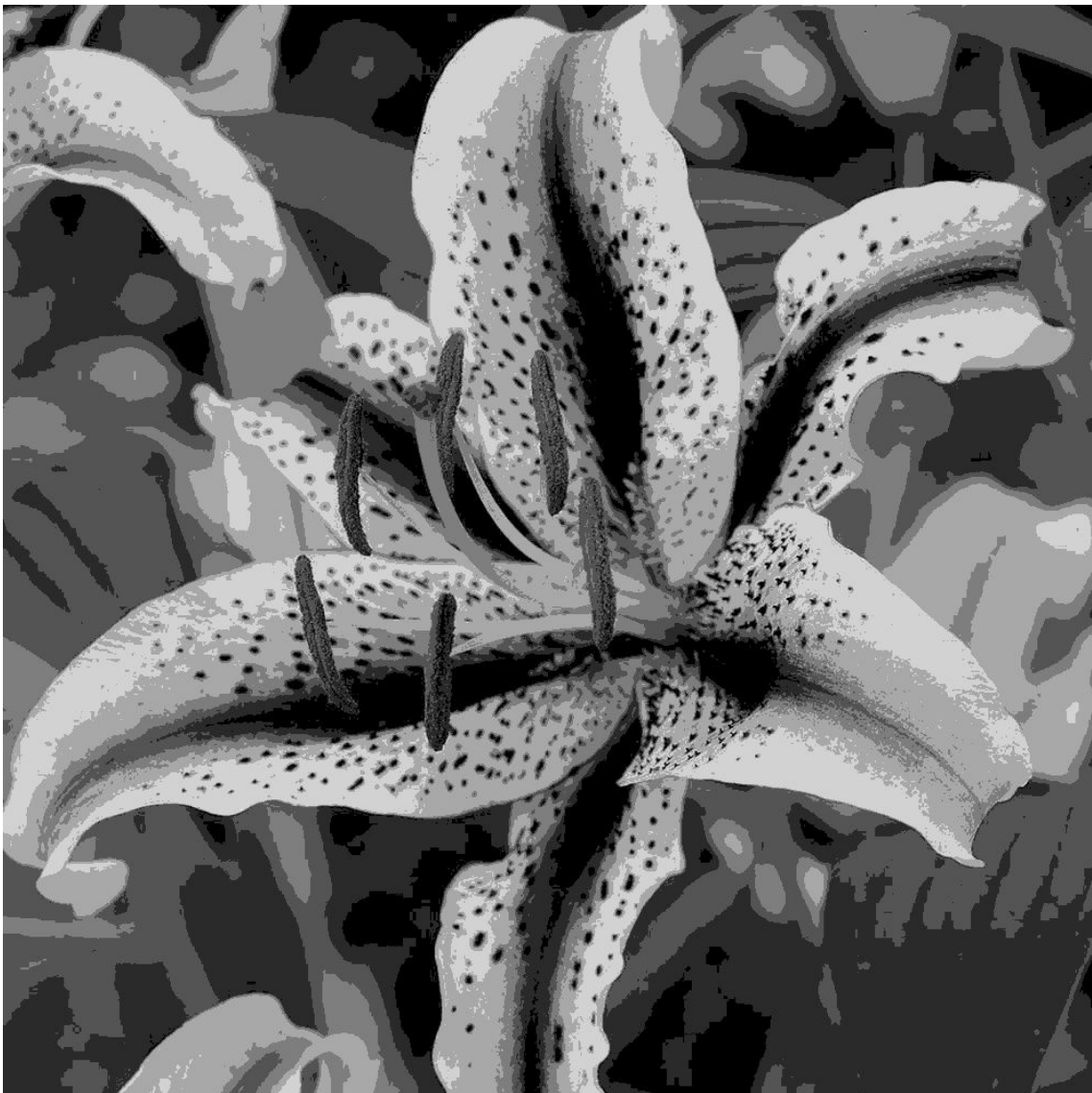
$$255 / 42 = 6,07...$$

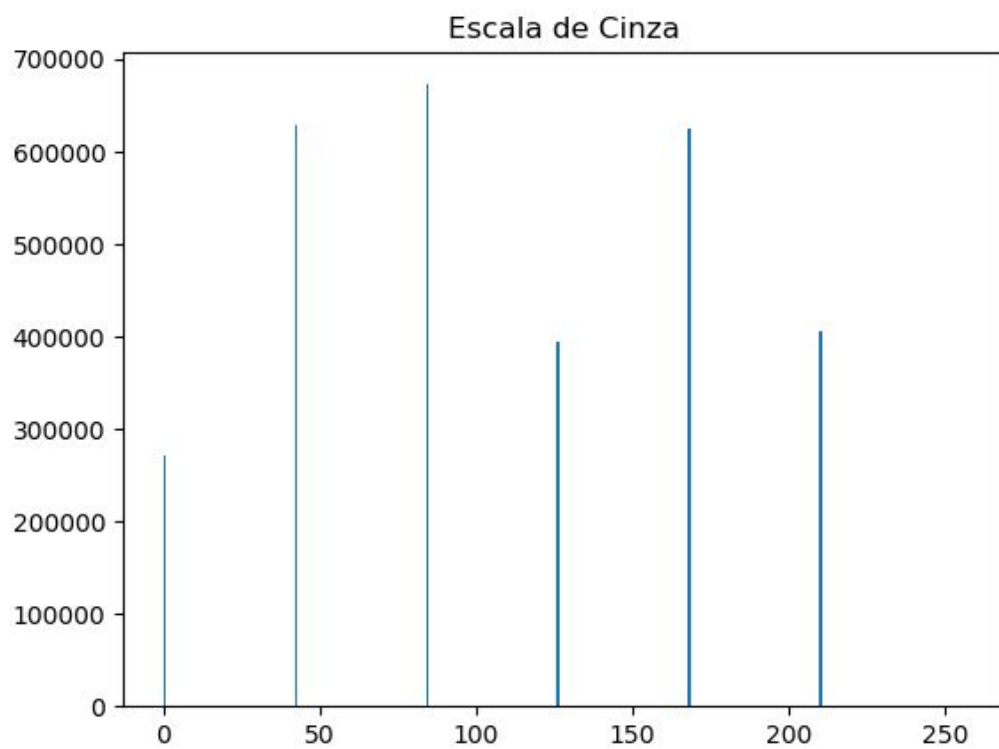
É utilizada uma função do NumPy para descartar os valores decimais.

Assim, teremos uma imagem com 6 tons de cinza. A conta é feita desta forma para descartar a parte decimal dos números e alterar o vetor para que possua apenas 6 valores possíveis.

Como resultado é aplicado uma média de cada um dos 6 tons de cinza que preenchem os demais tons removidos. Há uma maior perda de detalhes na imagem em relação a 8 tons de cinza.

Teste - 6 - 6 tons





3.3. Redução de tons de cinza - 2 tons

Neste exemplo o usuário selecionou a opção 2, que tem como função criar uma nova imagem com apenas 2 tons de cinza.

$255 / 127 = 2,00...$

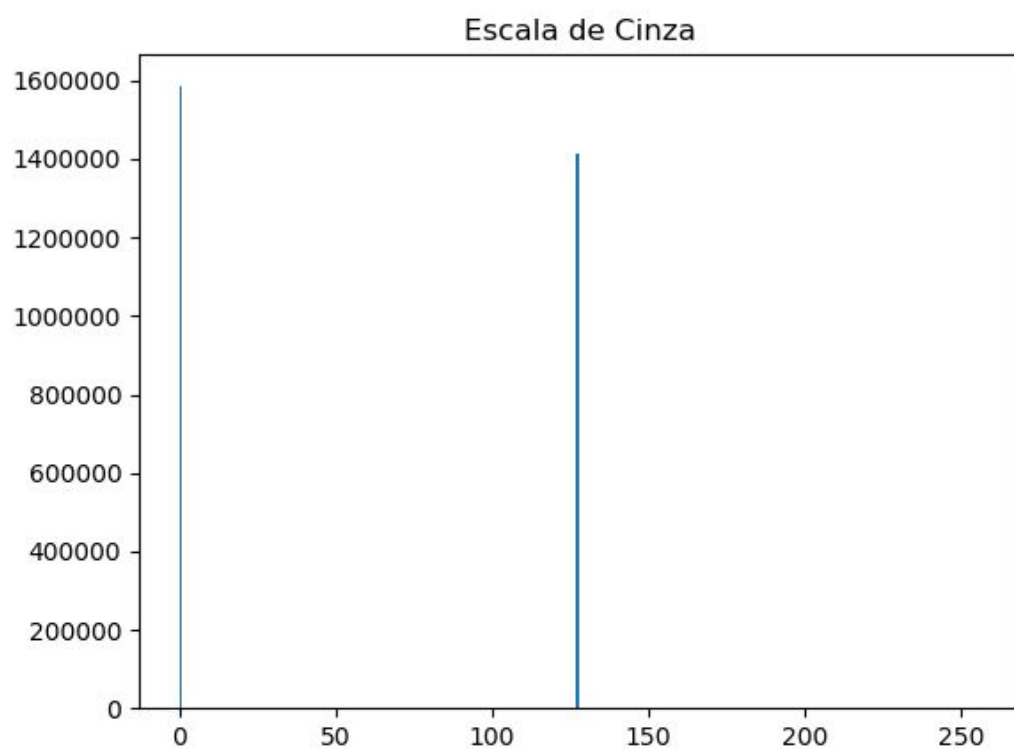
É utilizada uma função do NumPy para descartar os valores decimais.

Assim, teremos uma imagem com 2 tons de cinza. A conta é feita desta forma para descartar a parte decimal dos números e alterar o vetor para que possua apenas 2 valores possíveis.

Como resultado é aplicado uma média de cada um dos 2 tons de cinza que preenchem os demais tons removidos. Esses 2 tons variando de preto a uma média de valor entre preto e branco (cinza). Há perda quase total de detalhamento da imagem, conseguimos apenas diferenciar algumas pétalas.

Teste 7 - 2 tons

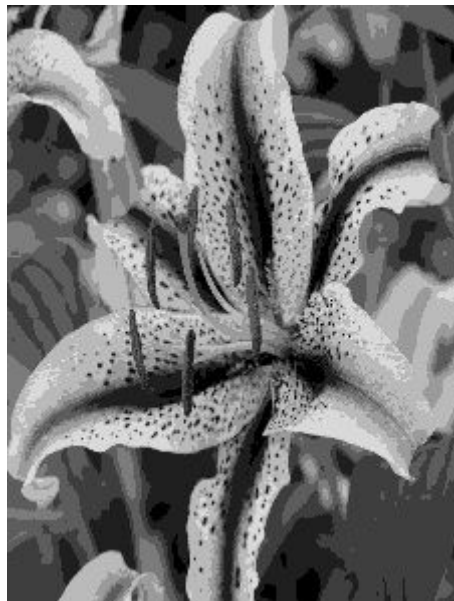


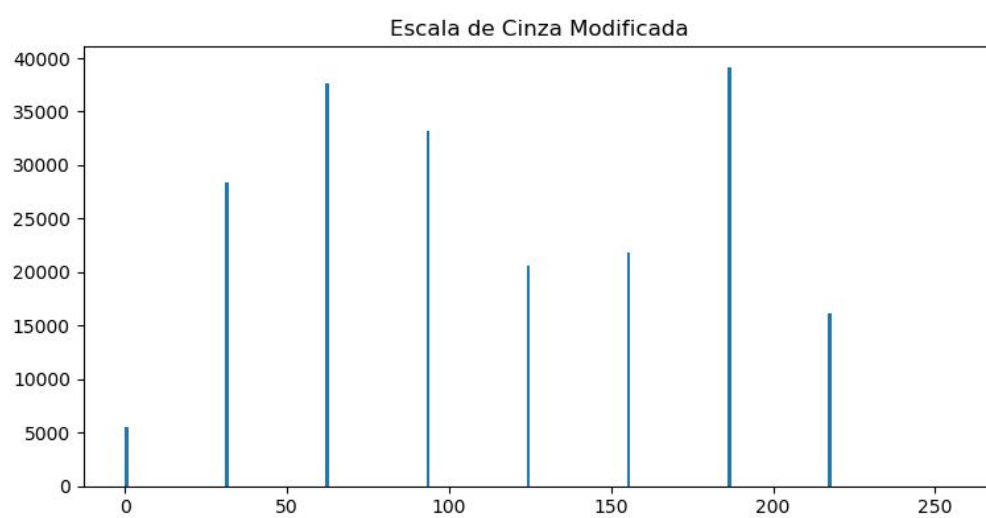
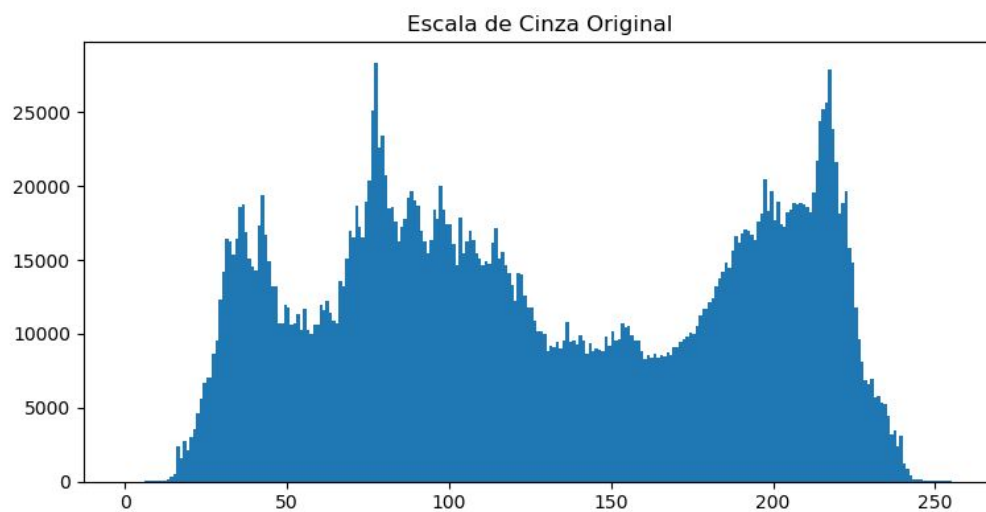


4. Aplicação de múltiplos efeitos

Foram aplicados os efeitos de Escalar (90%), Largura (450), Altura (600) e Redução de linhas e colunas $N = 2$, e foi reduzido para 8 tons de cinza, e é exibido ambas as imagens, além de seus histogramas.

Teste 8 - múltiplos efeitos aplicados





5. Repositório do programa:

Link: <https://github.com/CrazyAMK/Faculdade/tree/master/Processamento%20de%20imagens>