

Relatório - Trabalho 0

1. Introdução

O trabalho proposto consiste na realização de operações básicas com imagens (em formato PNG e em escala de cinza) utilizando a linguagem Python e suas diversas bibliotecas disponíveis:

- obter estatísticas de largura e altura (em pixels) e intensidades mínima, máxima e média;
- obter o histograma da imagem e salvá-lo em formato PNG;
- obter o negativo da imagem e salvá-lo em formato PNG;
- obter o resultado de uma transformação da imagem reescalando a faixa de intensidade de cinza de [0, 255] para [120, 160] e salvá-lo em formato PNG.

2. Execução, entradas e saídas

O código-fonte está contido inteiramente no arquivo *t0.py* e está implementado em Python 3.6.

Para executá-lo, é necessário colocar o caminho de uma imagem em PNG como parâmetro de execução. Por exemplo: `python t0.py 'D:/Google Drive/MC920/baboon.png'`

As estatísticas da imagem serão exibidas no console. Já as imagens transformadas serão salvas na pasta de execução com nomes “145980_1_histogram.png” para o histograma, “145980_2_negative.png” para o negativo e “145980_3_transformed.png” para a imagem transformada.

3. Descrição da solução

As quatro operações propostas (obter estatísticas, exportar histograma, exportar negativo e exportar imagem transformada) estão divididas em métodos para cada uma delas.

Para a leitura da imagem original, foi utilizado o método *imread* da biblioteca *SciPy*, que a armazena como um *ndarray*, tipo contido na biblioteca *NumPy*. As estatísticas da imagem foram extraídas diretamente por meio de operações desta mesma biblioteca.

A obtenção do histograma foi feita através da função *calcHist* da biblioteca *OpenCV*. O retorno dessa função foi utilizado diretamente em operações da biblioteca *matplotlib* para geração do gráfico e exportação como imagem PNG.

3.1. Geração do negativo da imagem

Para gerar o negativo da imagem original, foi utilizada basicamente uma operação de matrizes. Seja M a matriz que representa a imagem original e N a matriz que representa o negativo da imagem original.

$$\begin{bmatrix} 255 & 255 & \dots \\ \dots & \dots & \dots \\ \dots & \dots & 255 \end{bmatrix} - M = N$$

Com exceção da matriz preenchida inteiramente por números 255, que foi gerada pela função *full* da biblioteca *NumPy*, o cálculo foi utilizado com operações de matrizes oferecidas de forma nativa por Python.

A exportação para arquivo de imagem em formato PNG é feita utilizando a biblioteca *matplotlib*.

3.2. Transformação na faixa de intensidades de cinza

O cálculo da imagem transformada com uma nova faixa de intensidades de cinza foi realizado também utilizando operações com matrizes em Python.

Sejam *max* e *min* os valores máximo e mínimo, respectivamente, do novo intervalo de intensidades. Lembrando que o intervalo atual é de 0 a 255, portanto, tem 256 elementos.

Primeiramente, foi calculado um fator de conversão (*factor*) entre as faixas de intensidade:

$$factor = (max - min + 1) / 256$$

Posteriormente, uma matriz *newimage* representando a imagem transformada é preenchida de modo que cada pixel terá seu valor de cinza dentro do novo intervalo e mantendo a proporcionalidade entre os tons presentes na imagem original.

$$newimage[i, j] = (image[i, j] * factor) + min$$

Em Python, uma particularidade é poder realizar o cálculo acima sem a necessidade de iterar explicitamente dentro da matriz. Dessa forma, pode-se perceber no código que a linha correspondente a esse cálculo não está dentro de laços de repetição explícitos.

4. Testes e resultados obtidos

Foram realizados testes principalmente utilizando duas imagens:

- “*baboon.png*”, contida no conjunto de amostras fornecido inicialmente e presente no enunciado;
- “*DSC00114.png*”, uma foto retangular de acervo do próprio autor deste relatório que foi convertida de JPEG para PNG e posteriormente transformada em escala de cinza.

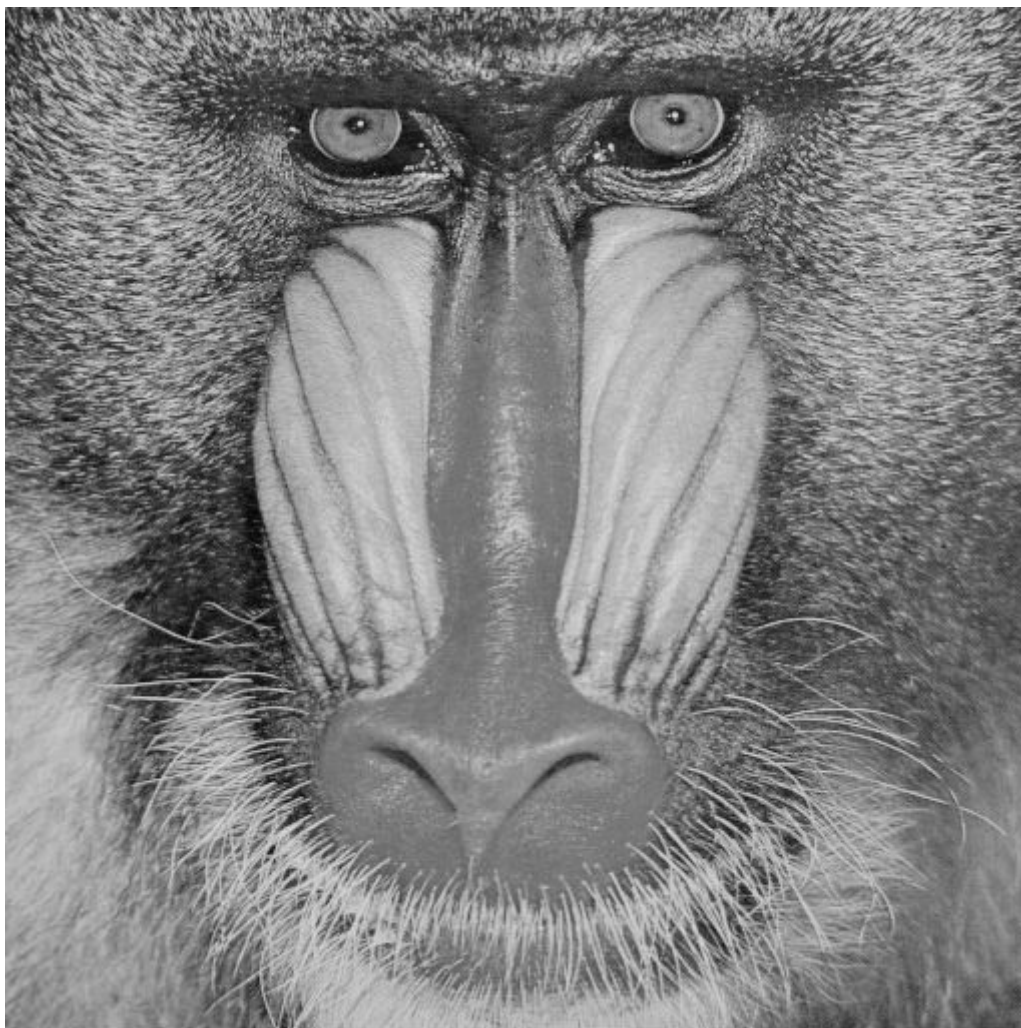


Figura 1 - “baboon.png” em tamanho original (512 x 512, em pixels)



Figura 2 - “DSC00114.png” reproduzida aqui em tamanho reduzido (tamanho original em pixels de 4608 x 3456)

O destaque a essas duas imagens se deve ao fato de terem sido fundamentais para balizar decisões tomadas durante a implementação da solução.

A imagem do primata foi utilizada largamente durante a implementação para comparação com os resultados esperados para ela presentes no enunciado do trabalho.

Já a imagem do ônibus teve importância no sentido de representar um passo maior na checagem do que fora implementado. Por ser retangular, permitiu checar se os valores apresentados como largura e altura pelo programa não haviam sido trocados. Outra característica é o tamanho consideravelmente maior em relação às amostras fornecidas. Isso propiciou uma análise de desempenho que levou a uma decisão de implementação que será explicada na seção 4.2.

4.1. Imagem “baboon.png”

A imagem do primata, como já foi explicado, foi utilizada para comparar as saídas do programa com os resultados esperados mostrados no enunciado do trabalho.

Não houve nenhuma grande dificuldade em utilizá-la. O único ponto de maior atenção foi a operação de fazer uma transformação mudando a faixa de intensidade de cinza. Inicialmente, pensou-se em simplesmente forçar os números menores que o menor da nova faixa para este, e os maiores que o maior da nova faixa para este. O resultado ficou diferente do esperado, pois foi violada a proporcionalidade entre as tonalidades de cinza presentes na imagem original.

Após a implementação de um algoritmo mantendo essa proporcionalidade foi obtido o resultado esperado.

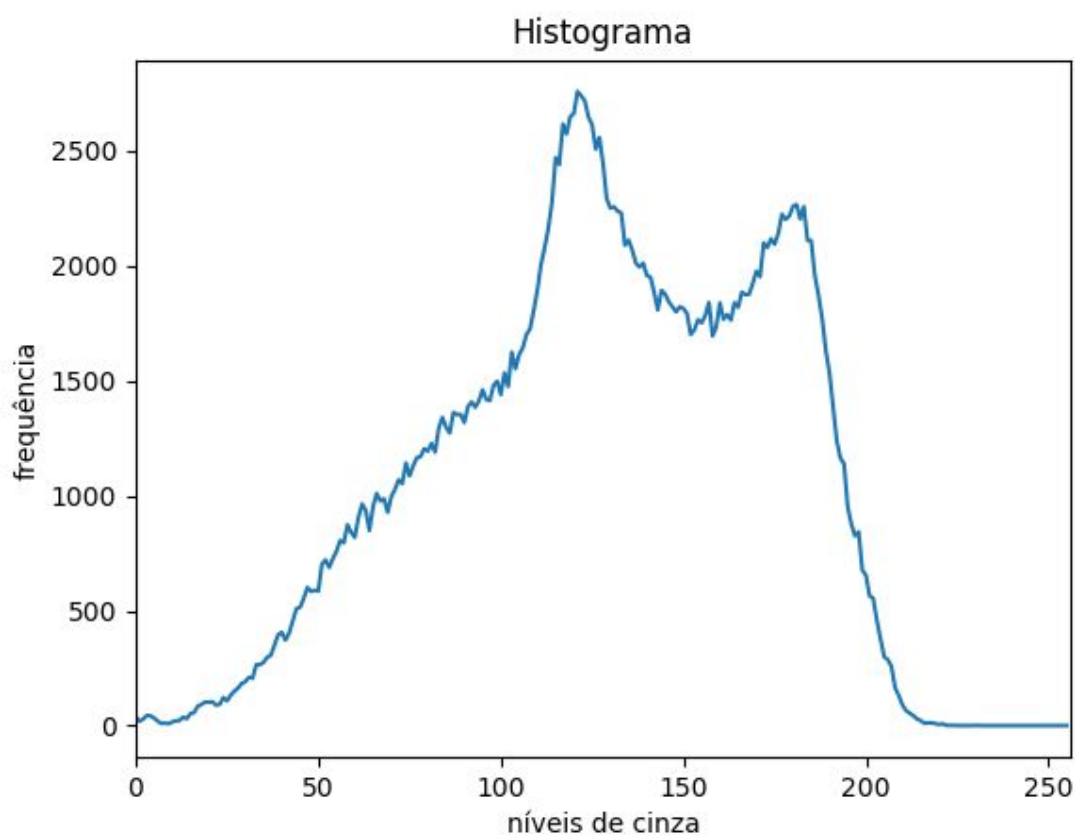


Figura 3 - Histograma obtido da imagem “baboon.png”

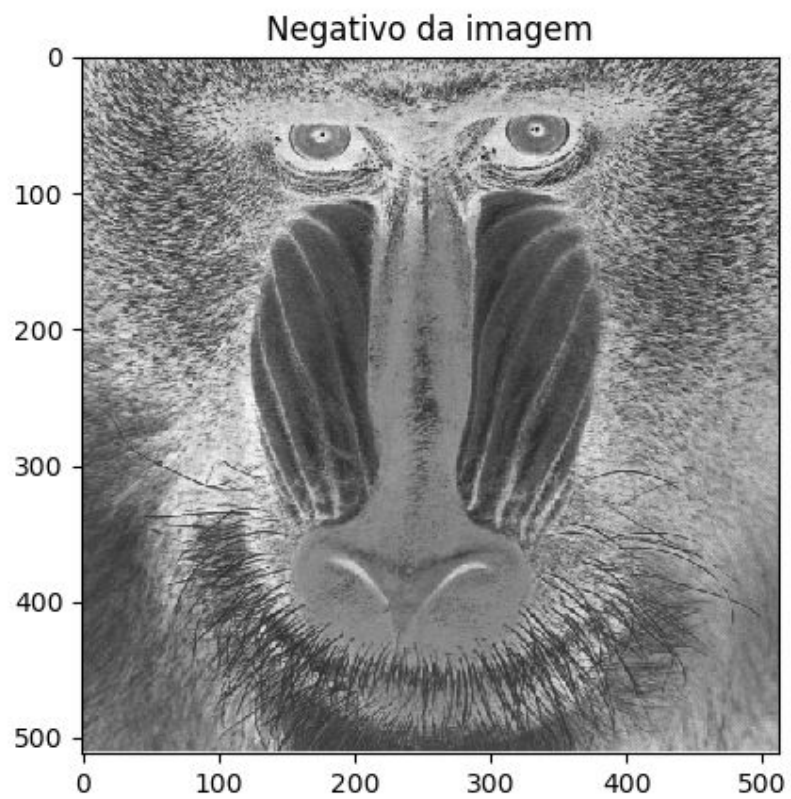


Figura 4 - Negativo obtido da imagem "baboon.png"

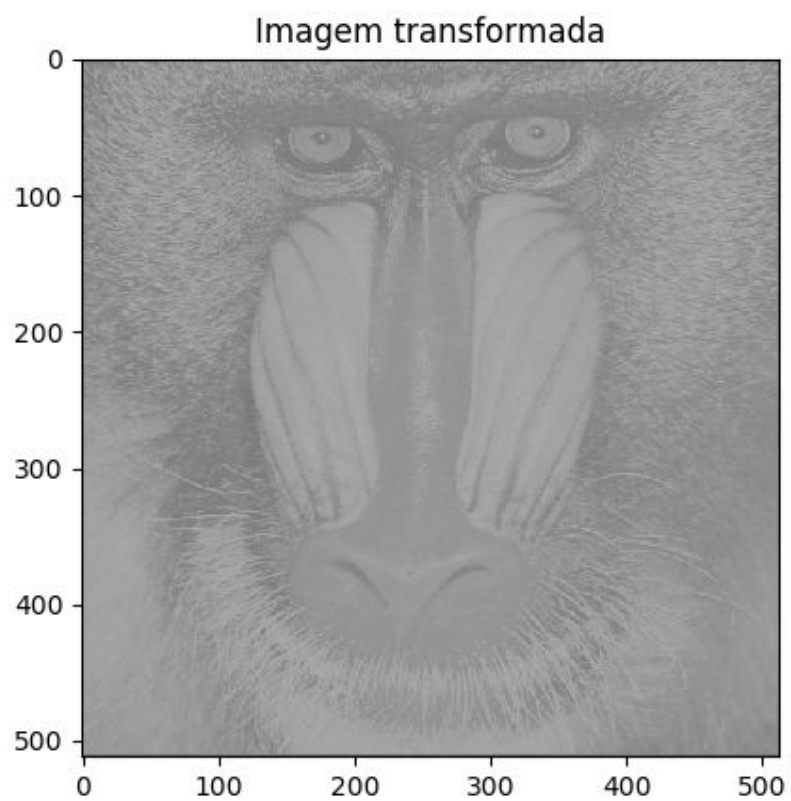


Figura 5 - Imagem transformada obtida de "baboon.png"

As estatísticas da imagem exibidas no console estão reproduzidas abaixo:

Largura: 512

Altura: 512

Intensidade mínima: 0.0

Intensidade máxima: 230.0

Intensidade média: 129.15

4.2. Imagem “DSC00114.png”

A imagem do ônibus, como foi dito, teve importância na verificação das estatísticas exibidas (pelo fato da imagem ser retangular) e em uma análise de desempenho da solução implementada.

Inicialmente, o primeiro desafio em utilizar esta imagem foi o fato de ela ser originalmente colorida. Foi convertida de JPEG para PNG e posteriormente foi convertida para escala de cinza por meio do software *PhotoFiltre* (disponível para Windows, utilizada versão gratuita limitada para uso pessoal). De qualquer forma, no carregamento da imagem, a operação *imread* é chamada utilizando um parâmetro opcional que força o carregamento das imagens como escala de cinza.

O programa implementado até então funcionava a contento, mas, com a imagem de dimensões maiores, tinha um tempo de execução consideravelmente maior.

Analisando o código, foi identificado que poderia ser otimizado se os métodos para cada operação com a imagem fossem executados de maneira assíncrona, o que era possível uma vez que nenhum deles mudava qualquer valor da imagem original carregada. Após uma rápida pesquisa, foi testado o código utilizando a biblioteca *asyncio*, obtendo uma melhora bastante perceptível.

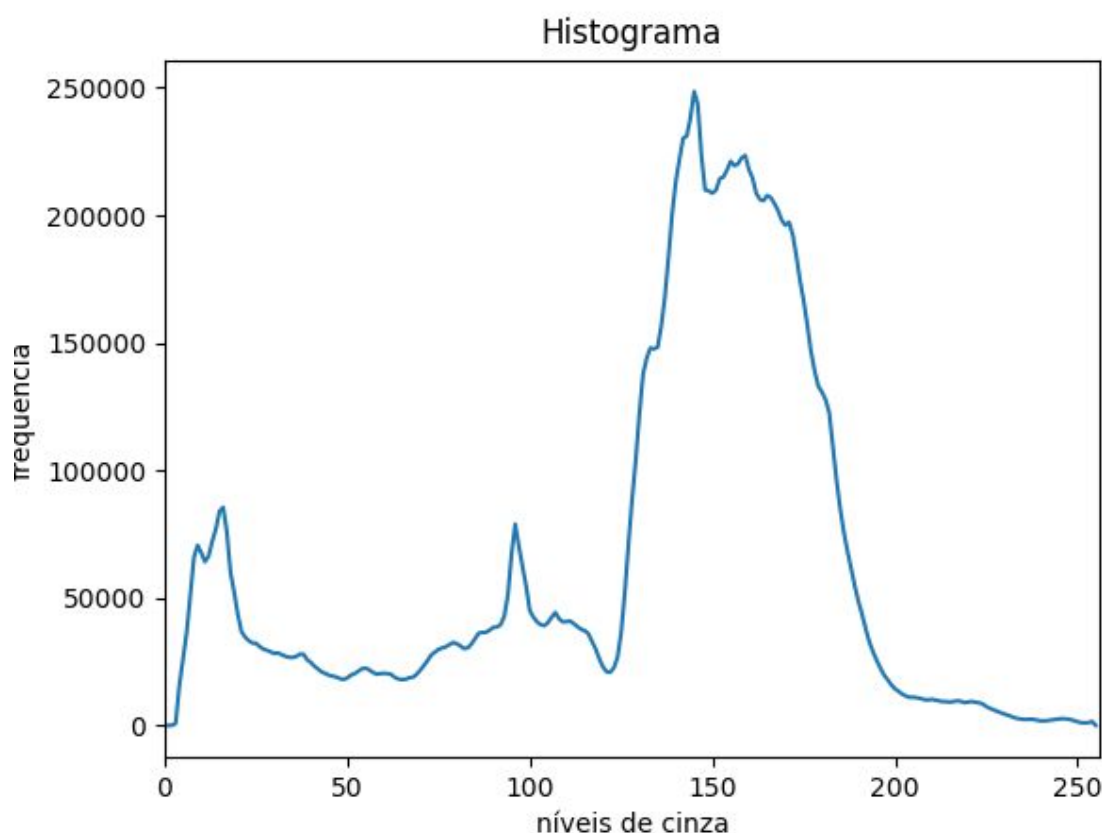


Figura 6 - Histograma obtido da imagem “DSC00114.png”

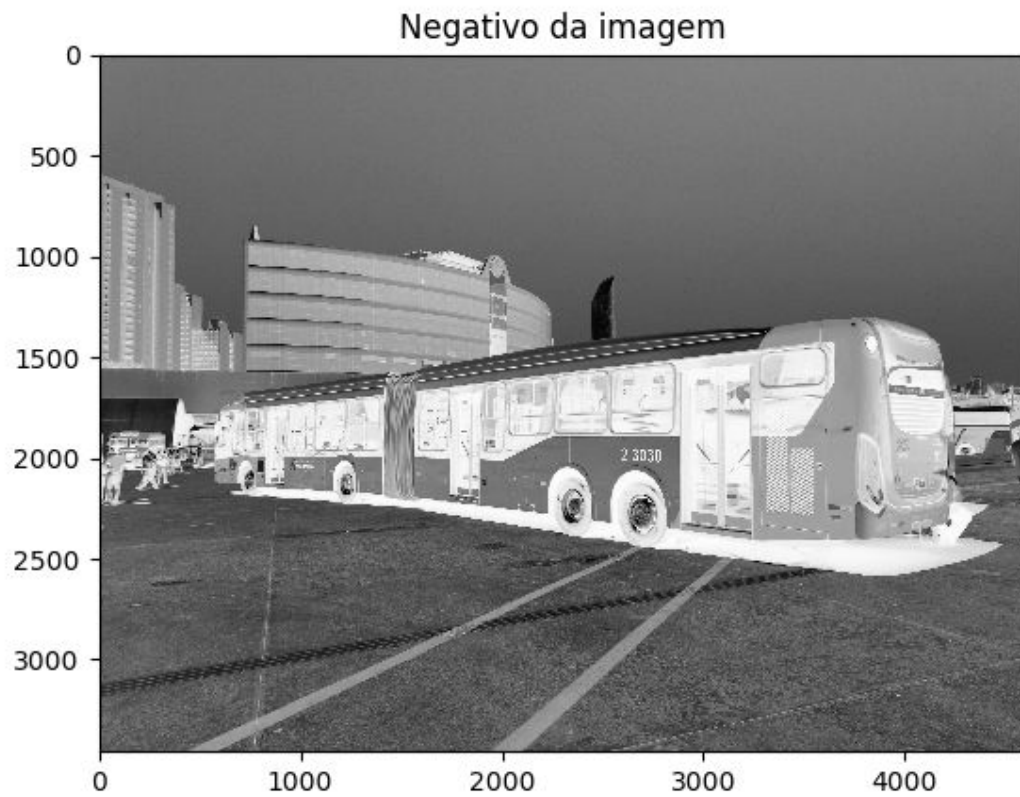


Figura 7 - Negativo obtido da imagem "DSC00114.png"

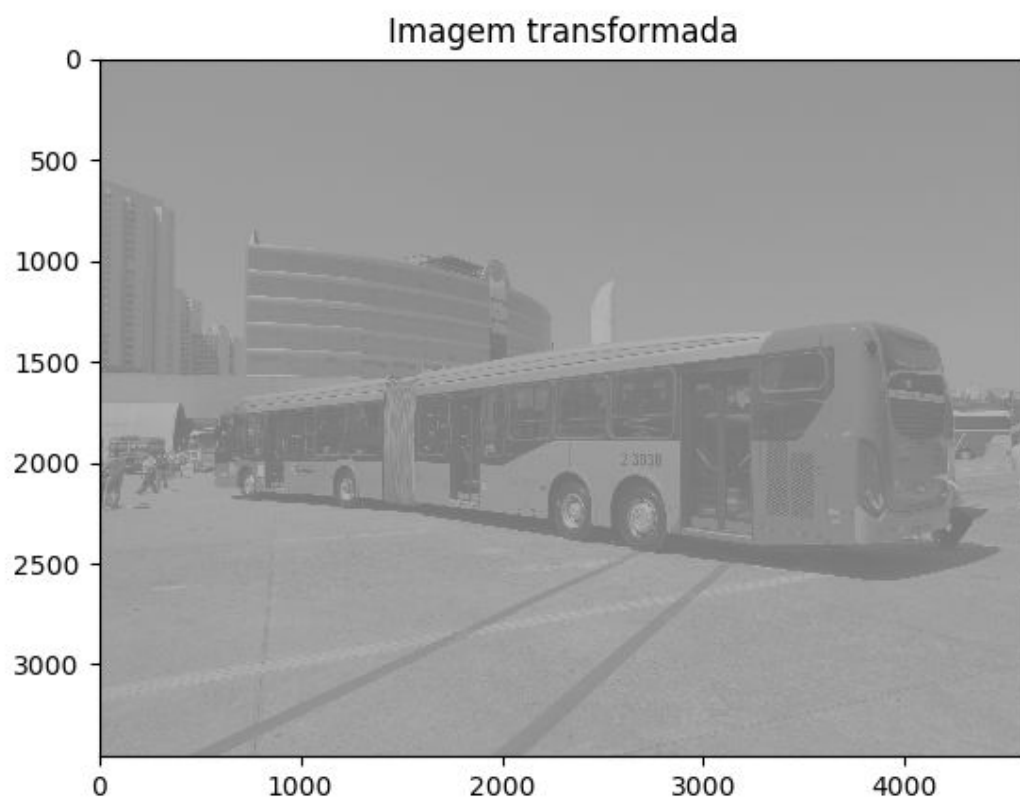


Figura 8 - Imagem transformada obtida de "DSC00114.png"

As estatísticas da imagem apresentadas via console foram as seguintes:

```
-----  
Largura: 4608  
Altura: 3456  
-----  
Intensidade mínima: 0.0  
Intensidade máxima: 255.0  
Intensidade média: 133.07  
-----
```

5. Problemas na implementação, defeitos existentes e outros aspectos

Não houve grandes problemas durante o desenvolvimento do trabalho proposto, nem mesmo quanto à configuração do ambiente em máquina própria.

Como foi dito na seção 4.1, houve um ponto de maior atenção quanto à operação de transformação para a nova faixa de intensidade de cinza pois a primeira tentativa foi de apenas forçar os números menores que o mínimo da nova faixa para este e os números maiores que o máximo da nova faixa para este. Após verificar que a imagem resultante não era compatível com a esperada (exibida no enunciado), foi adotada uma nova abordagem de manter a proporcionalidade entre os tons de cinza da imagem que foi bem sucedida.

6. Conclusão

Considera-se que o trabalho proposto tem sua importância no sentido de ajudar os estudantes a se familiarizar com operações básicas com imagens em Python, usufruindo das bibliotecas disponíveis.