



UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO

PROJETO 2 DE PROCESSAMENTO DIGITAL DE IMAGENS

Localização de objeto com variação de luminosidade

Tiago Lucas Leal, 760937

Lucas Martins de Almeida, 758983

São Carlos, SP, 2022

1 Resumo

Este trabalho tem como objetivo explicar e implementar uma técnica para a localização de um objeto com variação de luminosidades, a qual busca garantir que a localização do objeto ocorra com sucesso, mesmo caso a intensidade da luminosidade e do objeto a ser localizado sejam diferentes. Para isso foi desenvolvido um código em Python 3 com as bibliotecas Matplotlib, Numpy, Cv2 e Scipy. Foram realizados testes em imagens com diferentes valores de intensidade luminosa, uma mais clara e outra mais escura que a imagem original, para demonstrar o funcionamento correto da técnica de maneira prática.

2 Introdução

Em grande parte das ocorrências em que existe a necessidade de se encontrar um objeto numa imagem, tal objeto e o objeto na imagem não estão com a mesma intensidade luminosa. A técnica técnica de localização de objeto com variação de luminosidade é usada para resolver esse caso e tornar possível localizar um objeto em determinada imagem sem estarem com o mesmo nível de intensidade de luminosidade, por meio da utilização de diferenças quadráticas.

3 Objetivo

O objetivo desse trabalho é demonstrar o funcionamento da localização de objeto com variação de luminosidade através do desenvolvimento e execução de um algoritmo utilizando a linguagem Python 3 com as bibliotecas Matplotlib, Numpy, Cv2 e Scipy, que funcione com objetos em diferentes intensidades luminosas.

4 Metodologia

Seção da explicação da metodologia empregada para a implementação da técnica. Primeira parte da Teoria do método e a segunda parte para explicação do código implementado.

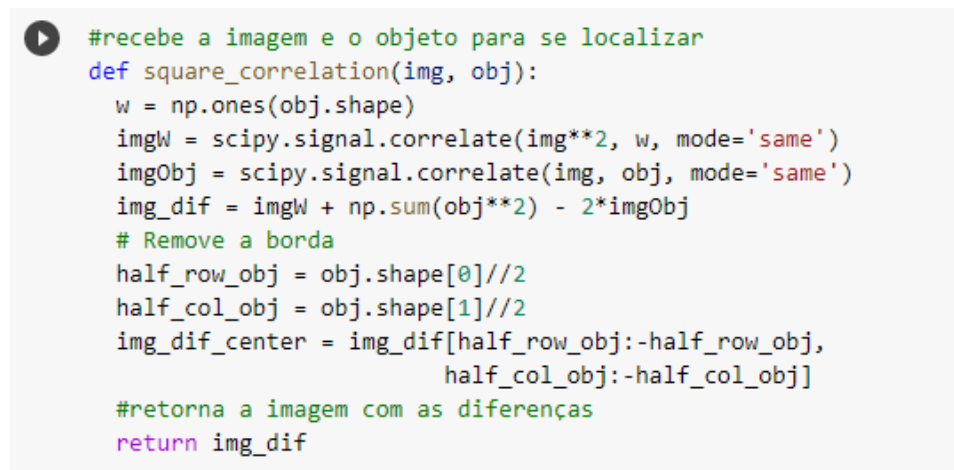
4.1 Teoria do Método

O procedimento da técnica utilizada para a localização de um objeto com variação de luminosidades é calcular a diferença quadrática entre duas imagens, uma sendo o objeto e a outra sendo a imagem no qual este será procurado, resultando, ao fim desse processo, em uma imagem diferença.

Em seguida, buscamos o menor valor para todas as imagens diferença, variando a intensidade da luminosidade(a transformação Gamma) e, quando o calculamos, representa que descobrimos o local em que o objeto está na imagem. Em seguida, desenhamos um retângulo ao redor para destacá-lo.

4.2 Explicação do Código

Os códigos mais importantes de nosso algoritmo são, representado na Figura 1, a função que calcula a diferença quadrática e, na figura 2, a função que calcula o menor valor resultado das diferenças quadráticas com diferentes graus de luminosidade.

A screenshot of a code editor showing a Python function named 'square_correlation'. The code is color-coded: comments are green, function definitions and return statements are purple, and other code is black. The function takes 'img' and 'obj' as arguments. It calculates the square correlation by first correlating the squared image with a ones matrix, then correlating the original image with the object, and finally combining them. It also removes the border by centering the result based on the object's dimensions.

```
#recebe a imagem e o objeto para se localizar
def square_correlation(img, obj):
    w = np.ones(obj.shape)
    imgW = scipy.signal.correlate(img**2, w, mode='same')
    imgObj = scipy.signal.correlate(img, obj, mode='same')
    img_dif = imgW + np.sum(obj**2) - 2*imgObj
    # Remove a borda
    half_row_obj = obj.shape[0]//2
    half_col_obj = obj.shape[1]//2
    img_dif_center = img_dif[half_row_obj:-half_row_obj,
                             half_col_obj:-half_col_obj]
    #retorna a imagem com as diferenças
    return img_dif
```

Figura 1: Código da função diferença quadrática em Python 3.0.

```
[ ] #procura a posição da matrix com menor valor
def find_minimum(img):
    num_rows, num_cols = img.shape
    menor_valor = img[0,0]
    indice_menor_valor = (0, 0)
    for row in range(num_rows):
        for col in range(num_cols):
            valor = img[row,col]
            if valor<=menor_valor:
                menor_valor = valor
                indice_menor_valor = (row, col)
    return menor_valor, indice_menor_valor
```

Figura 2: Código da função que calcula o menor valor em Python 3.0.

5 Resultados

A técnica foi aplicada em 1 imagem, com o objeto a ser encontrado sendo alvo de alterações na luminosidade, num caso sendo mais escuro e outro sendo clareado.

A imagem e o objeto a ser localizado, utilizados em suas versões originais. Nas figuras 3 e 4, respectivamente.

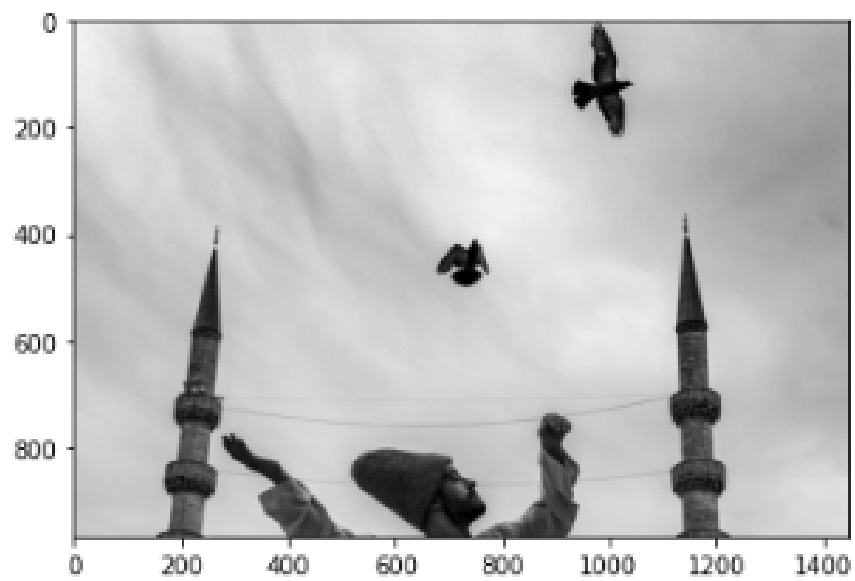


Figura 3: Imagem original.

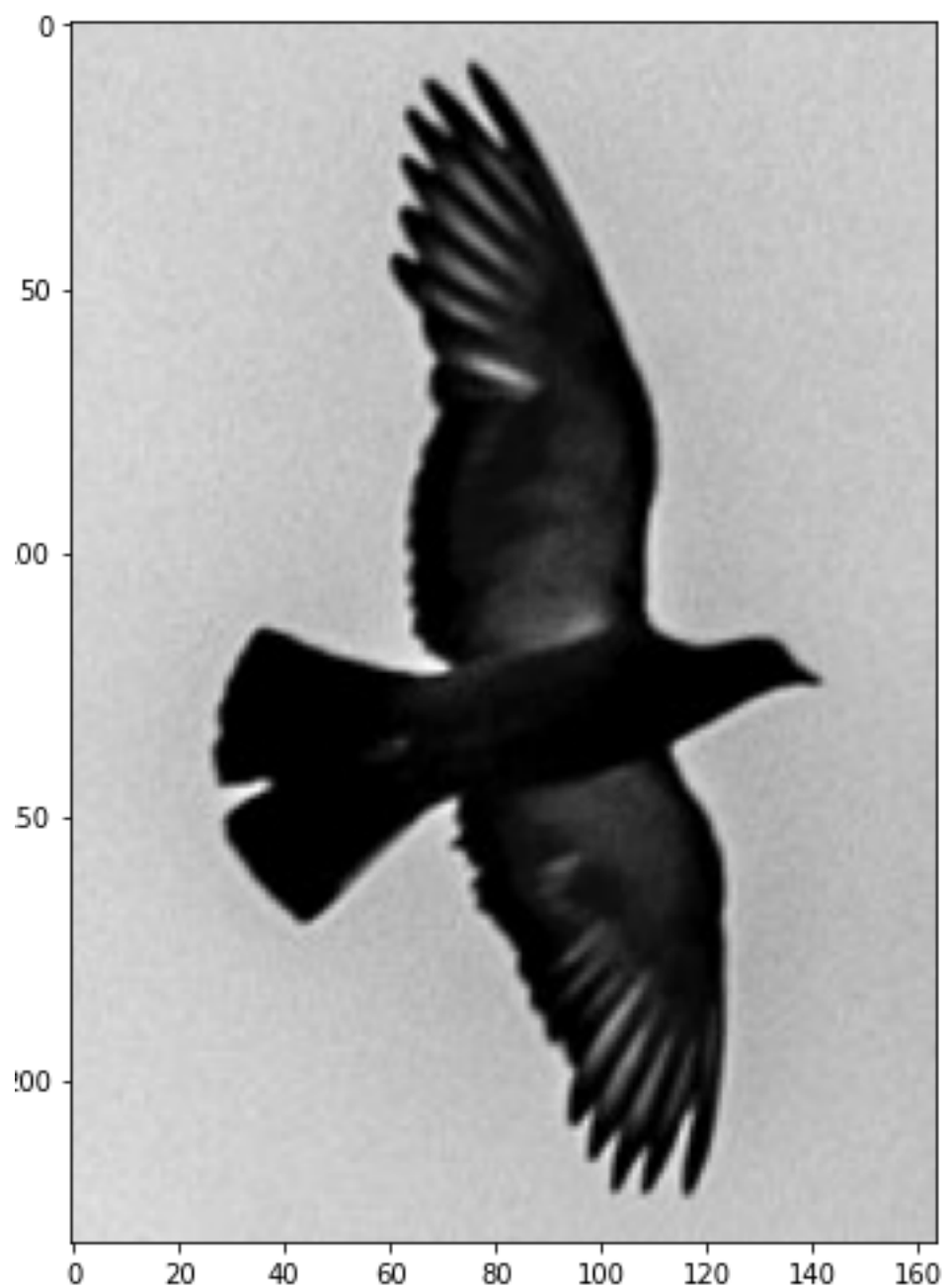


Figura 4: Objeto a ser localizado.

Em seguida, o resultado da busca com alteração de luminosidade escurendo a o objeto:

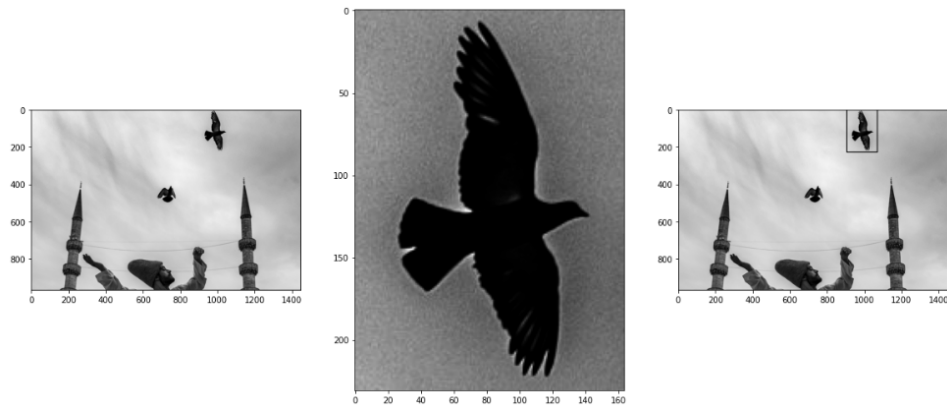


Figura 5: Resultado com objeto escuro.

logo após, o resultado da busca com alteração de luminosidade clareando a objeto:

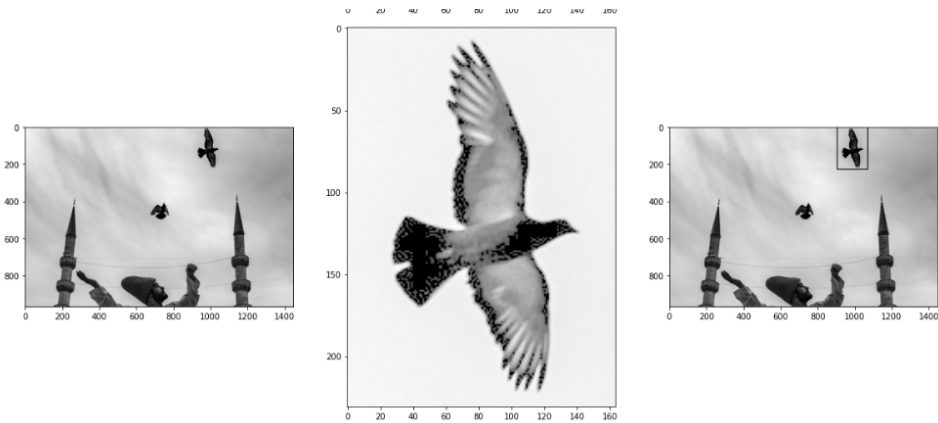


Figura 6: Resultado com objeto escuro.

6 Conclusão

Os resultados produzidos pela execução do algoritmo implementado foram iguais aos esperados e demonstraram a eficácia do mesmo, ao localizar a ave, independente da luminosidade da imagem que foi passada para busca. O trabalho, envolvendo o estudo, compreensão e desenvolvimento do algoritmo contribuiu bastante para a fixação do conhecimento dado em aula, principalmente sobre como funciona a localização de objetos e alterações de luminosidade.