Practical work N^{o}2

Objectives:

1. Use Octave commands related to image processing
2. Create Octave scripts for image manipulation

Commands

- **pkg load image** : import the image package
- **A=imread('name_file_image.extension')** : load an image into a matrix A
- **[m,n] =size(A)** : size of image A
- **imshow(A)** : display image A in a figure
- **imfinfo('name_file_image.extension')** : display image information
- **figure** : create a new figure
- **subplot(m,n,p)** : create a matrix of **m** by **n** graphs. The desired image is positioned at **p**. The subfigures are numbered from **1** to **mxn** from left to right then from top to bottom
- **imresize(A,** SCALE) **:** Modify the resolution of image by a factor SCALE
- **imwrite(B,'name.extension')** : save image B under name.extension
- **imrotate(A,ANGLE)** : rotation of image A by an angle ANGLE in degrees
- **isbw (A) :** returns **true**=1 if the image A is binary, otherwise 0
- **isgray(A) :** returns **true**=1 if the image A is grayscale
- **isind(A)** : returns **true**=1 if the image A is indexed
- **isrgb (A)** : returns **true**=1 if the image A is color
- **rgb2gray (A)** : convert color image A to gray level ∈ [0, 255]
  conversion formula: **0.2989 * R + 0.5870 * G + 0.1140 * B**
- **B=im2bw(A,S)** : convert image A into binary with threshold S ∈ [0,1] % B is logical
- **B= im2bw(A)** : convert image A into binary using **Otsu** method % B is logical
- **B=A>S :** convert image A (gray level) to binary with threshold S ∈ [0, 255] % B is logical
- **B= A>s1 & A<s2 :** binary image after multi-thresholding [s1,s2] % B is logical
- **im2double(A)** : convert A to double precision ∈ [0,1] (A is of type: Color, GS, BW)
- **im2uint8(A)** : the opposite of **im2double**
- **[B X] =rgb2ind(A) :** convert image A (color) to indexed image B (X is the color map)
- **A= Label2rgb(B,X)** : convert image B (indexed) to color image A.

Exercise 1

Work on Octave command window.

1. Load 2 images **naruto.jpg** and **cameraman.tif**
2. Display the sizes of the 2 images
3. Display both images in two separate figures
4. Display both images in one figure
5. Add titles for each image

Exercise 2

Write a script that allows to:

1. Load the image **Madagascar.jpg** in matrix A
2. Create the submatrix B of A (200:500,150:700, :)
3. Create an image C where A is rotated 45 degrees
4. Display A, B and C on the same figure
5. Save the image B on the hard disk

Exercise 3

Write a script that allows to:

1. Load an image A **cameraman.tif**
2. Increase the resolution of A to 200% (image B)
3. Compare A and B on two different figures
4. Reduce the resolution of A to 25% (image C)
3. Compare A and C on the same figure

- Can we restore the image A from C using **imresize**?

Exercise 4

Write an Octave script that:

1. Read the image A= **Madagascar.jpg**
2. Show in the same figure the image A and its negative B, as well as the curves of changing in gray level of line number 50 of A and B.

Exercise 5

Write an Octave script that allows to:

1. Read the image A= **naruto. jpg**
2. Show in the same figure A, the grayscale and the binary images of A.
3. Convert A to an indexed image
4. Restore A from the indexed image

Exercise 6

1. Write an Octave function **MODIFY(image, R,G,B)** that replaces the black color of an image with a color entered as an argument to this function (think of a solution without using loops).

2. Write a script that asks the user to read 3 values (RGB), then call **MODIFY** on the image **formes.jpg** (use the following color: R=210 G=50 B=90)

Extra exercises

Exercise 1

Write a function **Rotation( )** which rotates an image according to an angle entered as an argument (use the following formulas).

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\sin\theta + y\cos\theta \end{cases}$$

Exercise 2 (Exam 2022)

Let A and B two binary images (same dimensions).

Without using loops, write an Octave script which creates image C (color) from A and B. The colors of image C are: **background**: green, **star**: red, **ellipse**: blue.

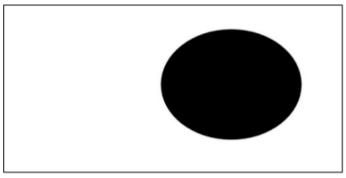


Image A

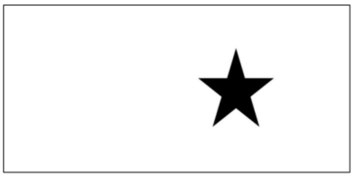


Image B

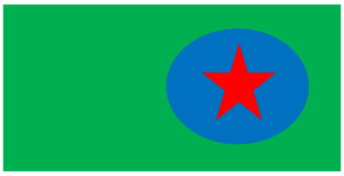


Image C

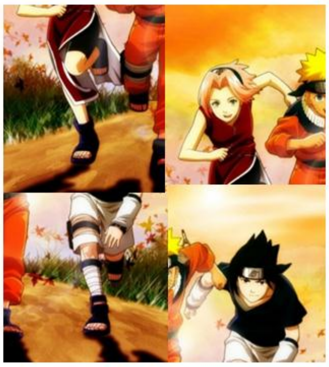
Exercise 3

We want to create a game that allows to sort 4 blocks of an image (256x256) into the appropriate places (1,2, 3, 4) in order to form **Naruto** image (512x512)

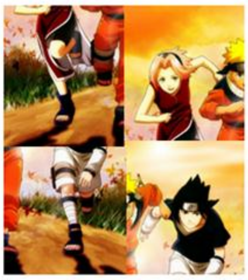
Steps :

1. Load the **Naruto .jpg** image (512x512)
2. Create 4 images A B C D for each region 1, 2, 3 and 4 of size (256x256) each
3. Randomly merge A, B, C and D to form an unsorted image
4. Each time ask the user to choose the zones to swap (example 1 with 4) and display the final image until the sorted image is obtained
5. The swap operation stops when the user obtains the sorted image.

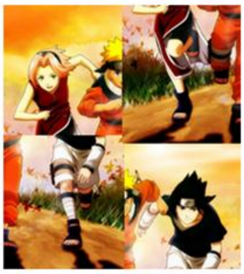
1	2
3	4



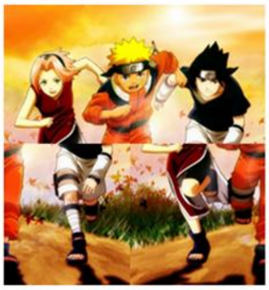
Example



```
what are the zones to swap
1
2
```



```
what are the zones to swap
2
4
```

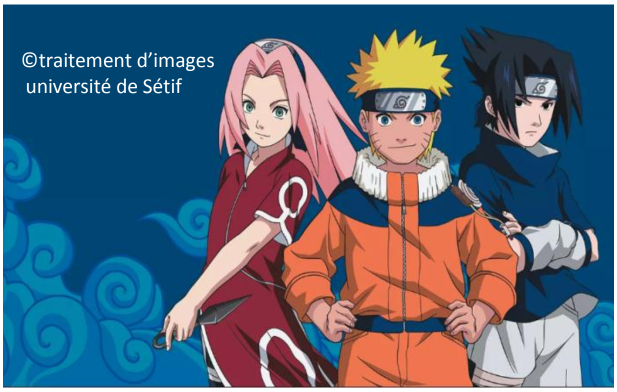


```
what are the zones to swap
3
4
```

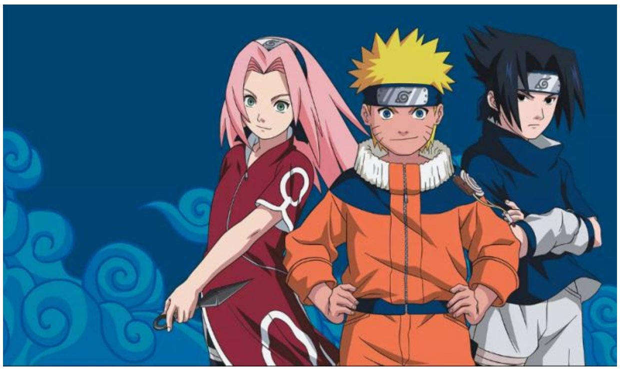
Exercise 4 (Exam 2019)

1. Write an Octave function **DELETE(x1, x2, y1, y2, originalimage)** which erases a text included in a color image. The function should replace all pixels in the rectangle**(x1,x2,y1,y2)** with the average value of the pixels in the line above the rectangle (limited by **y1 and y2**).
2. Call **DELETE** in a script (use the image Naruto2.jpg as input).

Original Image ***(Naruto2.jpg)***



final Image



Bibliographic references

- **Downloadable books**

1. *Pillow: Image Processing with Python. Kindle Edition. Michael Driscoll 2021*

2. *Fundamentals of Digital Image Processing. A Practical Approach with Examples in Matlab*, C. Solomon, T. Breckon,Wiley-Blackwell

3. *Digital Image Processing Using MATLAB, 2nd Edition*, Rafael C. González, Richard Eugene Woods, Steven L. Eddins

4. *Digital Image Processing, 4rd Edition by Rafael C. Gonzalez*, Richard E. Woods 2022

5. *Practical Image and Video Processing Using MATLAB, 1 edition*. Hoboken, O. Marques, N.J: Wiley-IEEE Press, 2011.

6. *Signal Processing and Performance Analysis for Imaging Systems*, by S. Susan Young and Ronald G. Driggers, 2008

7. *An Interdisciplinary Introduction to Image Processing: Pixels, Numbers, and Programs (The MIT Press) by Steven L. Tanimoto (Author)*