

## Module : TP Vision Artificielle

### TP N° : 01

## Initiation au traitement d'images



### Objectif de ce TP:

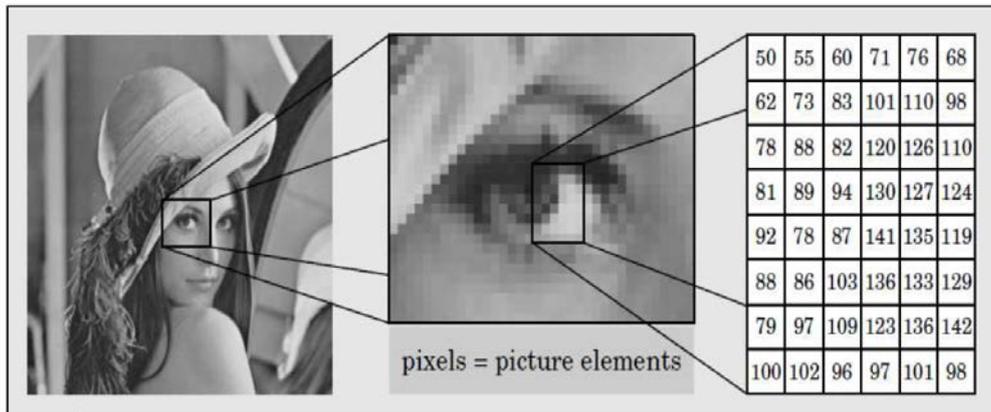
À la fin de ce TP, l'étudiant sera capable de :

- Découvrir la toolbox "Image processing" de Matlab,
- Maitriser les commandes de base de manipulation d'images sous Matlab,
- Distinguer les différents types d'images,
- Savoir les caractéristiques et les propriétés des images numériques.

## Définitions:

### 1. Image numérique:

Une image numérique est un ensemble fini de pixels codés par des bits issus d'un échantillonnage spatial. Généralement chaque pixel est codé sur 8 bits (0 à 255) pour les images N&B. Pour les images couleurs, chaque pixel possède trois composantes (RVB) sur 8 bits, ce vecteur caractérise l'intensité de chaque couleur dans l'ensemble RVB (rouge, vert, bleu). L'image est alors une matrice de nombre.



### 2. Principe de l'histogramme:

Pour afficher l'histogramme d'une image en niveaux de gris, il est nécessaire de calculer le nombre de pixels pour chaque niveau de gris. Une fonction, qui permettra de réaliser et afficher l'histogramme, doit être créée. Le résultat sera comparé à l'histogramme réalisé sous Matlab grâce à la fonction `imhist()`.

### 3. Principe de la quantification:

La quantification permet de coder une image, qui est initialement en 256 niveaux de gris, en  $n$  niveaux de gris où  $n$  est un paramètre qui peut être modifié par l'utilisateur lorsqu'il le souhaite. On découpe les niveaux de gris en intervalles de même amplitude.

## Travaux à réaliser (Script Matlab):

### Manipulation 1:

1. Ouverture et lecture d'une image: A l'aide du Matlab, lire l'image "coloredChips.png" puis l'afficher.

➤ Code Matlab

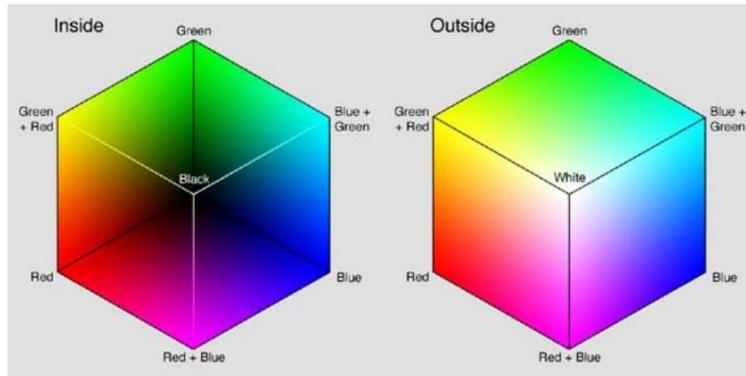
```
clc;
close all;
clear all;
im=imread('coloredChips.png');
imshow(im)
```

2. Pour connaître les propriétés de cette image, on tape sur Matlab:

- whos `im`
- size(`im`)

### Manipulation 2:

1. Décomposition de l'image en trois plans de couleur R, G, B.



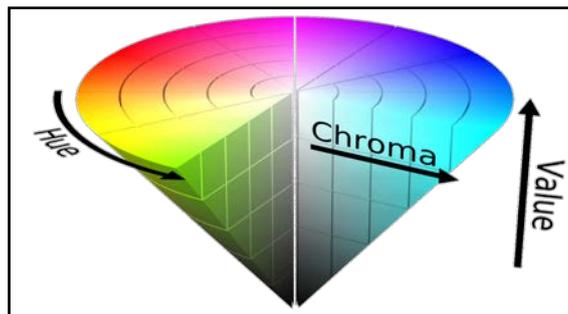
- Code Matlab

```
clc;
close all;
clear all;
im=imread('coloredChips.png');
r=im; g=im; b=im;
r(:,:, [2 3])=0;
g(:,:, [1 3])=0;
b(:,:, [1 2])=0;
figure
subplot(221); imshow(im); title('image originale');
subplot(222); imshow(r); title('la composante rouge');
subplot(223); imshow(g); title('la composante verte');
subplot(224); imshow(b); title('la composante blue');
```

### Manipulation 3: La transformation HSV et en niveau de gris (grayscale)

La signification de l'espace de couleurs HSV est la suivante :

1. H (**hue**) : la longueur d'onde la plus dominante.
2. S (**saturation**) : la « pureté » d'une couleur ( la quantité du blanc dans la couleur).
3. V (**variance**) la valeur de luminance – brightness.



- A. Décomposez une image RGB en un ensemble de composante de l'espace HSV, comme dans la manipulation 2 et affichez-les. (utilisez `rgb2hsv`)
- B. Transformez une image RGB en niveau de gris et l'affichez, puis tracez l'histogramme correspondant. (utilisez `rgb2gray` et `imhist`)

#### Manipulation 4 : La quantification:

1. Quantifiez l'image en  $N = 8, 4$  et 2 niveaux de gris.
2. Que remarquez vous (en terme de qualité, résolution et taille) ?

➤ Code Matlab

```

clc;
close all;
clear all;
im=imread('coloredChips.png');
X=rgb2gray(im);
X=double(X)/255;
d8=256/8; I8=floor(X*255/d8)/(8-1);
d4=256/4; I4=floor(X*255/d4)/(4-1);
d2=256/2; I2=floor(X*255/d2)/(2-1);
figure
subplot(2,5,1); imshow(im); title('image RGB');
subplot(2,5,2); imshow(X); title('256 niveaux de gris');
subplot(2,5,3); imshow(I8); title('8 niveaux de gris');
subplot(2,5,4); imshow(I4); title('4 niveaux de gris');
subplot(2,5,5); imshow(I2); title('2 niveaux de gris');
% histogramme de chacune
hr=imhist(im(:,:,1)); hv=imhist(im(:,:,2)); hb=imhist(im(:,:,3));
subplot(2,5,6); plot([hr hv hb]); title('histogramme image RGB');
subplot(2,5,7); imhist(X); title('hist 256 niveaux');
subplot(2,5,8); imhist(I8); title('hist 8 niveaux');
subplot(2,5,9); imhist(I4); title('hist 4 niveaux');
subplot(2,5,10); imhist(I2); title('hist 2 niveaux');

```

3. On peut sauvegarder les nouvelles images en utilisant la commande suivante:

➤ Code Matlab

```
imwrite(I2, 'mon_image_binaire.png');
```

4. Faites la binarisation d'une image en niveau de gris par deux méthodes différentes

4.1. Par la commande `im2bw`.

4.2. Par les instructions `for ... end` et `if...else...end`.

5. Quel est le rôle de ces commandes: `imrotate`, `imresize`, `imcrop`.