



---

**Travaux pratiques : Les bases de traitement d'images**

**Master 2ME2S**

**Pr : M. Sayyouri**

**TP1 : Toolbox de Matlab de traitement d'images et de la vidéo**

Une opération typique de traitement d'images avec Matlab Image processing Toolbox se compose de plusieurs étapes. Dans ce TP, on va voir les opérations de lecture, affichage et stockage des images. Maîtriser ces opérations est important parce qu'on les utilise avec toutes les opérations de traitement d'images qu'on va étudier dans les autres TPs de ce module.

**Etape 1 : Lire et afficher une image**

Pour lire une image, nous utilisons la commande `imread( )`. Cette fonction lit une image à partir d'un fichier image et la stocke dans un tableau (matrice)

Exemple :

```
I = imread('lena.jpg');
```

`Imread` va deviner le format à partir de son nom (extension). Cette fonction reconnaît plusieurs formats. Voir la documentation de `imread`.

La plupart des formats de fichier d'image utilisent 8 bits pour stocker les valeurs des pixels. Quand ils sont lus dans la mémoire, Matlab les stocke comme des valeurs de type `uint8`. Pour les formats de fichiers prenant en charge les données 16 bits, Matlab les stocke avec le type `uint16`.

On peut aussi lire des images couleurs RGB.

Exemple:

```
RGB = imread('lena_color.bmp');
```

On peut aussi lire une image indexée avec sa colormap associée dans deux variables séparées :

```
[X, map] = imread('lena_color.bmp');
```

Matlab prend en charge plusieurs formats de fichiers graphiques, tels que HDF et TIFF pouvant contenir plusieurs images. Lorsqu'un fichier contient plusieurs images qui sont liées d'une certaine manière, comme une séquence temporelle, nous pouvons stocker les images sous forme de tableau 4-D. Toutes les images doivent être de la même taille. Par exemple, lorsqu'on utilise des fichiers TIFF, nous pouvons utiliser une valeur d'un index avec `imread` pour identifier l'image dans le fichier. Nous pouvons également utiliser `imfinfo` pour déterminer combien d'images sont stockées

dans le fichier.

Exemple :

Dans cet exemple, on va lire une série de 27 images à partir d'un fichier TIFF et les stocker dans un tableau 4D.

```
mri = uint8(zeros(128,128,1,27)); % réserver un tableau 4D
for frame=1:27
    [mri(:,:,,frame),map] = imread('mri.tif',frame);
end
```

## **Etape 2 : vérifier comment l'image apparaît dans le Workspace**

Matlab peut stocker des images sous diverse formes (uint8, uint16, doubles). Pour voir comment la fonction imread stocke les données de l'image dans le workspace, on peut le vérifier avec le navigateur du workspace.

Vérifier la représentation dans le workspace de Matlab des images des exemples précédents.

## **Etape 3 : Affichage des images**

Pour afficher une image, on utilise la fonction : imshow

(On peut utiliser l'outil intool qui est un environnement graphique intégré qui permet l'affichage des images et d'effectuer certaines tâches courantes de traitement d'image)

Exemple:

```
imshow(I)
```

On peut utiliser directement : imshow('image.bmp'), mais dans ce cas l'image n'est pas enregistrée dans le workspace.

La fonction image() affiche un objet image. Cette fonction crée un objet graphique de type image en interprétant chaque élément d'une matrice comme un index dans la colormap de la figure ou directement en tant que valeurs RGB, selon les données spécifiées.

La fonction impixel() renvoie les valeurs de couleur rouge, verte et bleue des pixels d'image spécifiés.

**Etape 4: faire le traitement (ce que on va faire dans les prochains TPs).**

**Etape 5: écrire (sauvegarder) l'image dans un fichier (sur disque)**

Pour écrire (sauvegarder) les images traitées ou les résultats du traitement dans un fichier, nous utilisons la fonction imwrite.

Exemple: `imwrite (I, 'barara_1.jpg');`

Dans ce cas, vous pouvez vérifier le résultat sur le disque. On peut aussi utiliser : `imfinfo('barara_1.jpg')`

La fonction imfinfo renvoie des informations sur l'image dans le fichier, telles que son format, sa taille, sa largeur et sa hauteur, etc.

`size(I)` par contre affiche la taille de l'image. Tester aussi `whos I`.  
Pour changer le format d'une image seulement, utilisez `imread()` puis  
utilisez `imwrite()`:

```
barbara_2= imread('barbara_1.jpg','jpg');  
imwrite(barbara_2,'barbara_2.tif','tif');
```