COMPUTER VISION

Rappel

- Citer quelques applications de la vision par ordinateur
- Quelle est la différence entre Image Processing et Computer Vision?
- Qu’est-ce qu’une image?

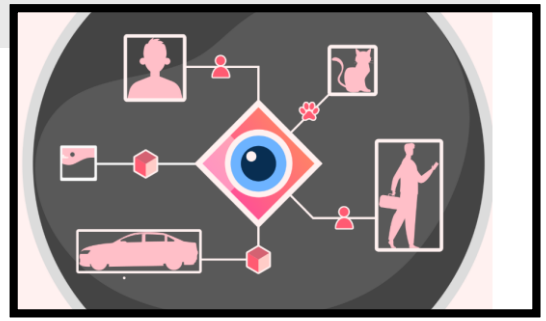
Reconnaissance faciale : Identifier et reconnaître des visages dans des images ou des vidéos, utilisé dans la sécurité, la surveillance et la gestion des identités.

Diagnostic médical : Analyse d'images médicales telles que les radiographies, les IRM et les scanners CT pour détecter les maladies, les lésions et les anomalies.

Navigation autonome : Permettre aux véhicules autonomes, tels que les voitures, les drones et les robots, de percevoir et d'interpréter leur environnement pour naviguer en toute sécurité.

Reconnaissance optique de caractères (OCR) : Convertir automatiquement du texte à partir d'images ou de documents numérisés en texte éditable, utilisé dans la numérisation de documents et la lecture automatisée de formulaires.

Surveillance vidéo : Analyser des flux vidéo en temps réel pour détecter des événements, des comportements ou des objets suspects, utilisé dans la sécurité publique, la surveillance de la circulation et la sécurité des bâtiments.



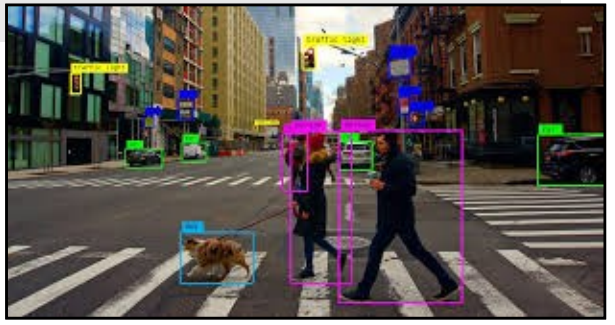
Interactions homme-machine : Permettre aux utilisateurs d'interagir avec des systèmes informatiques à l'aide de gestes, de mouvements ou d'expressions faciales, utilisé dans les interfaces utilisateur naturelles et la réalité augmentée.

Reconnaissance d'objets : Identifier et classer automatiquement des objets dans des images ou des vidéos, utilisé dans la recherche visuelle en ligne, le tri automatisé et la robotique.

Analyse de scènes : Extraire des informations à partir d'images pour comprendre la structure, la disposition et le contexte d'une scène, utilisé dans la cartographie, la modélisation 3D et la réalité virtuelle.

Contrôle de la qualité : Évaluer la qualité des produits en inspectant visuellement les défauts ou les irrégularités sur les surfaces, utilisé dans la fabrication industrielle et l'inspection des produits.

Indexation et recherche d'images : Organiser et rechercher des images en fonction de leur contenu visuel plutôt que de métadonnées textuelles, utilisé dans les moteurs de recherche d'images et les applications de gestion de photos.

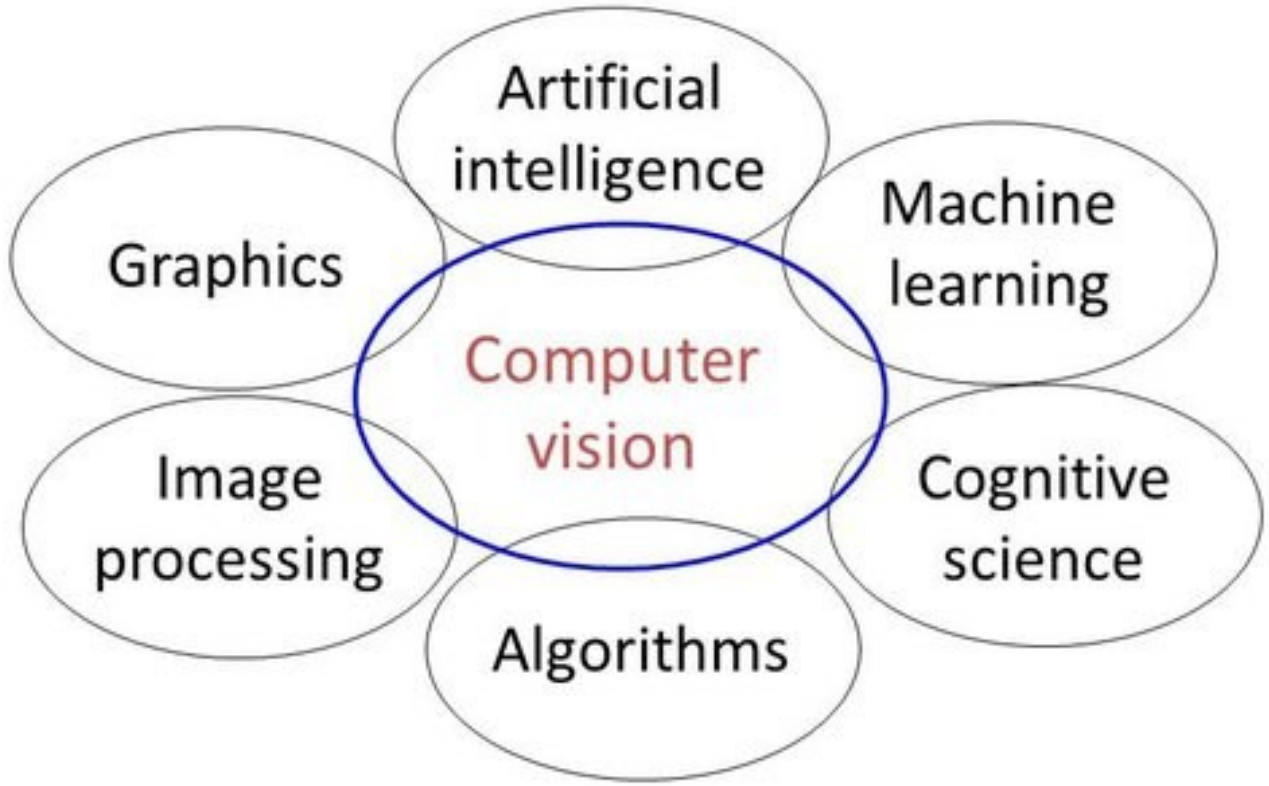


Rappel

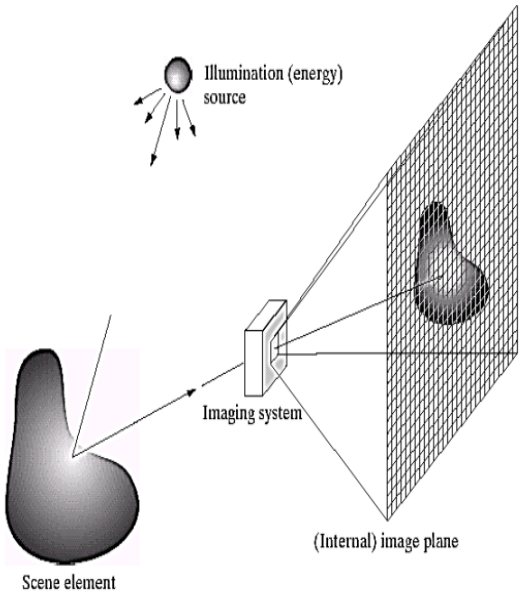
- La **vision par ordinateur** se concentre sur la compréhension et **l'interprétation** des contenus visuels par les ordinateurs, en permettant aux machines de "voir" et de comprendre leur environnement de la même manière que le fait un être humain.

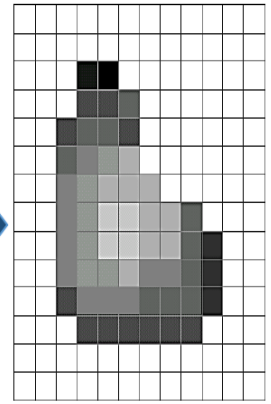
- Le **traitement d'images**, quant à lui, se concentre principalement sur la manipulation des images pour améliorer leur qualité, extraire des informations spécifiques ou effectuer des opérations telles que la restauration, la segmentation ou la détection de contours.

Related disciplines



Rappel





255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	20	0	255	255	255	255	255	255	255
255	255	255	75	75	75	255	255	255	255	255	255
255	255	75	95	95	75	255	255	255	255	255	255
255	255	96	127	145	175	255	255	255	255	255	255
255	255	127	145	175	175	175	255	255	255	255	255
255	255	127	145	200	200	175	175	95	255	255	255
255	255	127	145	200	200	175	175	95	47	255	255
255	255	127	145	145	175	127	127	95	47	255	255
255	255	74	127	127	127	95	95	95	47	255	255
255	255	255	74	74	74	74	74	74	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255

0=noir
255=blanc

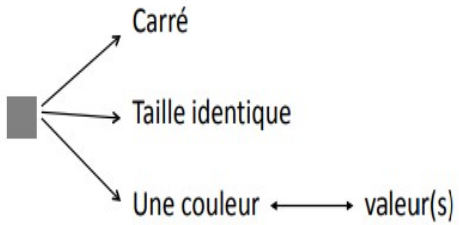
Une image, fondamentalement, est la représentation visuelle d'un **objet illuminé** par une **source de lumière**. Elle capture les valeurs qui reflètent l'intensité lumineuse émanant de l'objet. Ces valeurs varient en fonction de la quantité de lumière réfléchie par l'objet : une réflexion intense se traduit par des valeurs élevées, tandis qu'une absorption de lumière se traduit par des valeurs faibles. En d'autres termes, une image est un ensemble de **valeurs de pixels qui révèlent l'intensité lumineuse d'une scène ou d'un objet**, nécessitant une interprétation pour en comprendre le sens et les détails.

Rappel

- Une image matricielle est composée de pixel.

- Pixel est la contraction de PIcture Element.

- Le pixel est la plus petite unité de surface d'une image.



Une image est essentiellement une **matrice de pixels,** où chaque pixel détient une valeur représentative de **son niveau de luminosité ou de couleur**. Ces valeurs peuvent varier en fonction du type d'image : on peut **avoir une image en niveaux de gris** où chaque pixel est représenté par une seule valeur, **une image en couleur** où chaque pixel est composé de valeurs pour les canaux de couleur (**rouge, vert, bleu**), ou même **une image en fausses couleurs** où les valeurs ne correspondent pas nécessairement à des couleurs réelles mais sont utilisées pour représenter des informations spécifiques.

En résumé, une image est une structure de données organisée en **une grille de pixels**, chaque pixel portant des **informations visuelles** qui, lorsqu'interprétées, forment une représentation visuelle de l'objet ou de la scène capturée

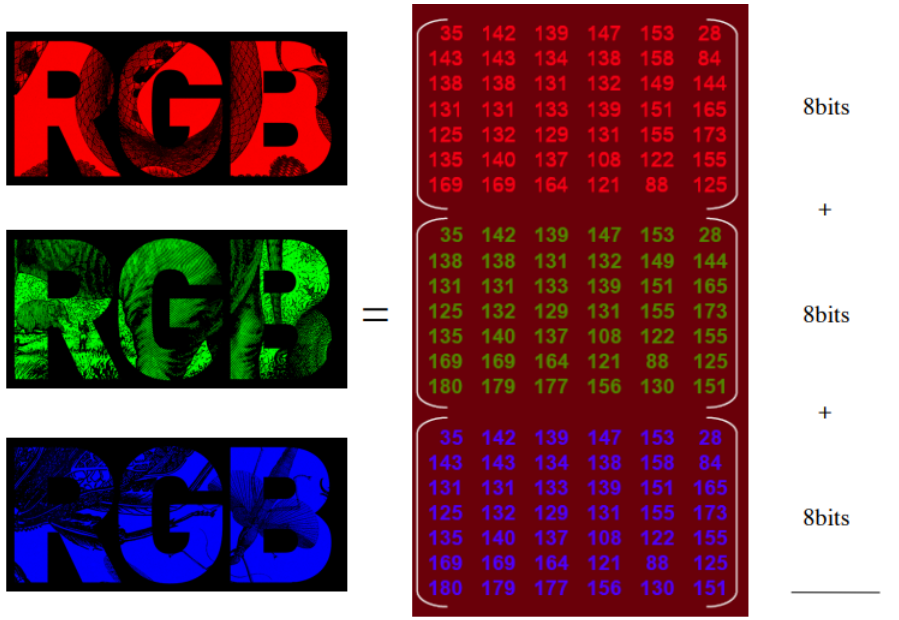
Rappel

Une **image RGB** est un type d'image numérique où chaque pixel est représenté par trois valeurs correspondant aux composantes de couleur rouge (R), verte (G) et bleue (B). Ces trois composantes primaires de couleur sont mélangées en différentes proportions pour former une vaste gamme de couleurs. Chaque composante de couleur peut généralement varier sur une échelle de 0 à 255, où 0 représente l'absence de cette couleur et 255 représente la pleine intensité de cette couleur. En mélangeant différentes intensités de rouge, de vert et de bleu, on peut créer une multitude de couleurs différentes.

Image couleur = codage RGB

Une image couleur corresponds à une superposition de 3 plans :

Rouge / Vert / Bleu (RGB = Red / Green / Blue)



24 bits => 2^{24} = 16 777 216 couleurs (TrueColor)

Rappel

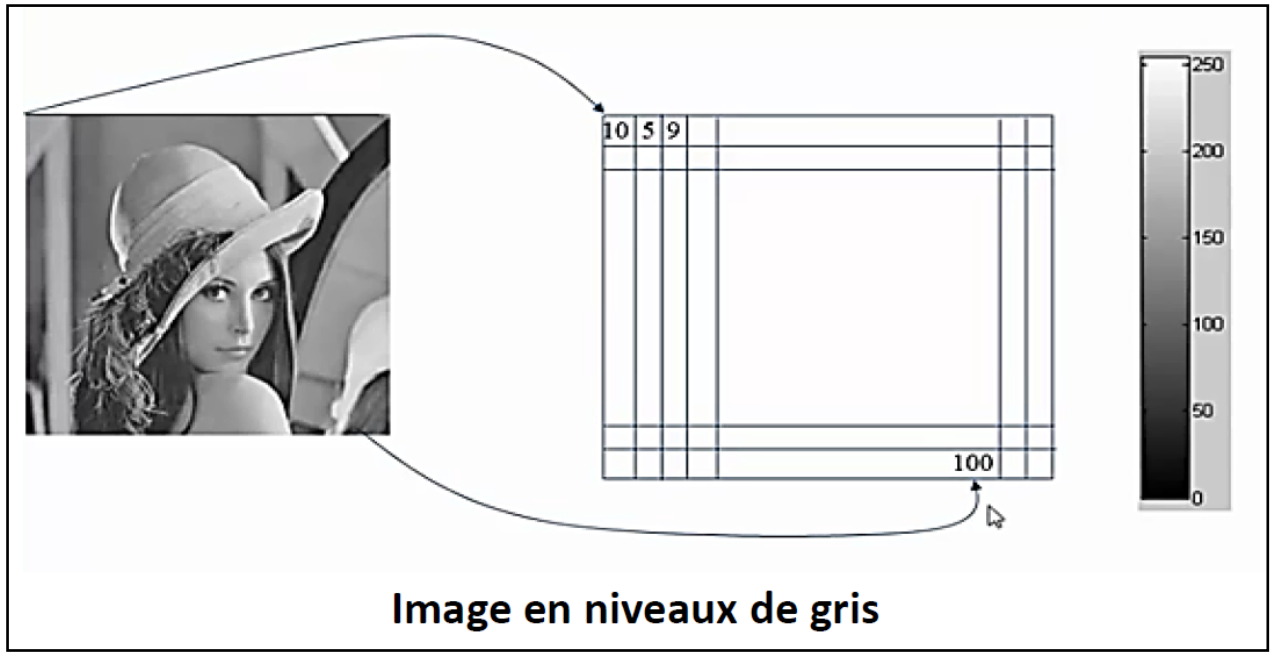


Image en niveaux de gris

Une **image en niveaux de gris** est un type d'image où chaque pixel est représenté par une seule valeur de luminosité, généralement sur une échelle de 0 à 255. Cette valeur indique l'intensité lumineuse du pixel, où 0 représente le noir absolu et 255 représente le blanc absolu. Les nuances de gris entre le noir et le blanc sont utilisées pour représenter différents niveaux de luminosité, créant ainsi une image monochrome. Ce type d'image est couramment utilisé dans divers domaines, notamment en imagerie médicale, en traitement d'images et en photographie artistique pour mettre en valeur les détails et les contrastes sans la distraction des couleurs

Rappel

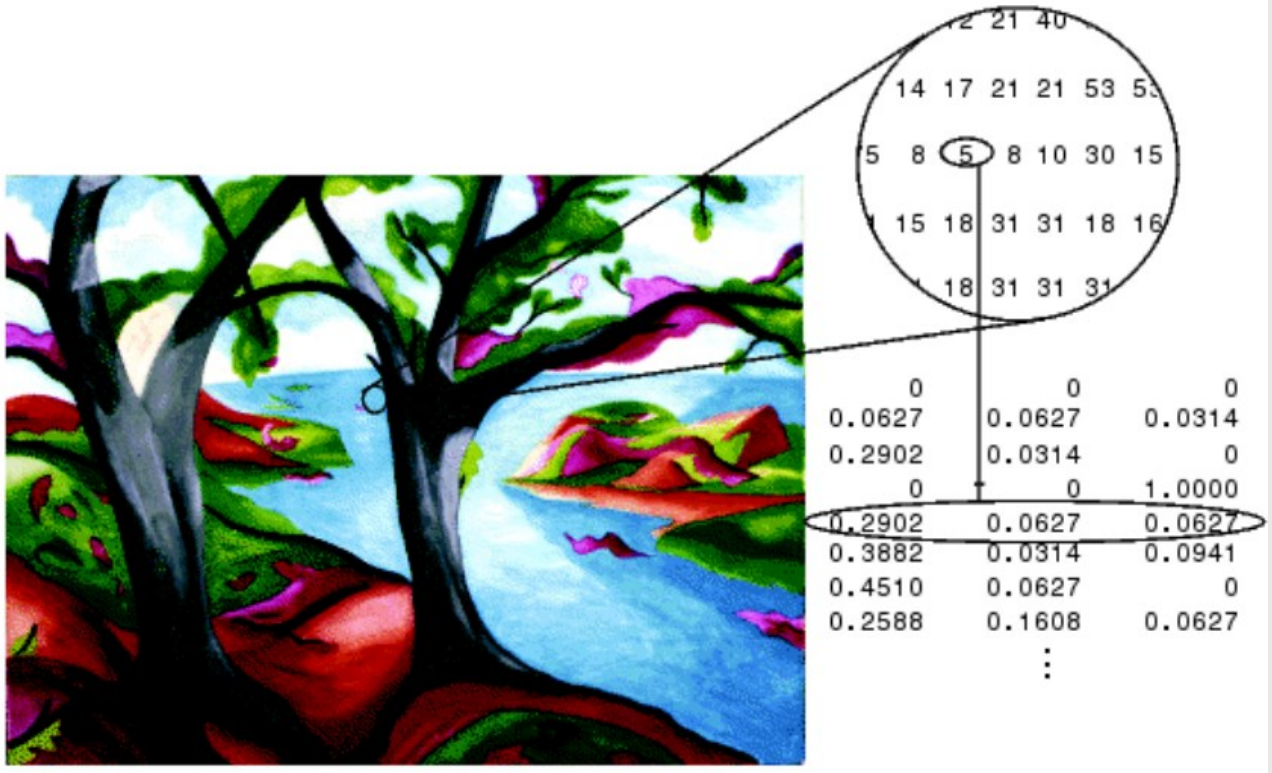


Image Courtesy of Susan Cohen

Une **image indexée** est un type spécifique d'image numérique où les couleurs sont sélectionnées à partir d'une **palette prédéfinie**, plutôt que d'être représentées par des valeurs RVB (rouge, vert, bleu) pour chaque pixel individuel. Au lieu de cela, chaque pixel de l'image est associé à un index correspondant à une couleur dans une table de couleurs prédéfinie, également appelée palette de couleurs ou table de couleur.

Cette approche permet de réduire l'espace mémoire nécessaire pour stocker l'image, car les informations sur les couleurs sont stockées de manière compacte dans la palette de couleurs plutôt que pour chaque pixel individuel. Les images indexées sont souvent utilisées dans des contextes où la taille du fichier est un facteur critique, comme sur le web ou dans des systèmes avec des ressources limitées

Image Processing avec MATLAB

Matlab, acronyme pour "**Matrix Laboratory**", est un langage de programmation et un environnement de développement largement utilisé dans les domaines de l'ingénierie, des mathématiques et des sciences.

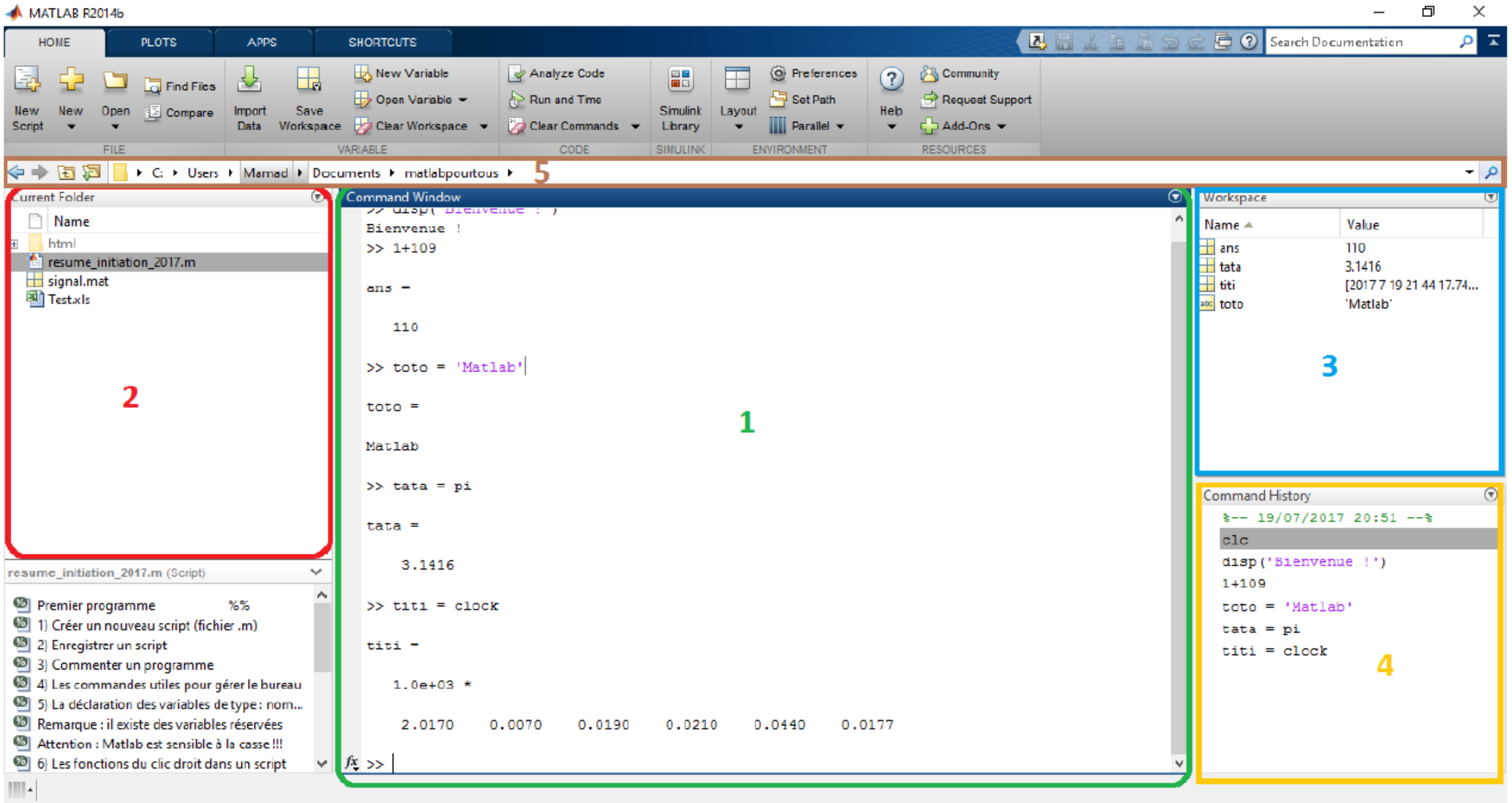
Voici quelques caractéristiques principales de Matlab :

- **Langage à typage dynamique** : Les variables n'ont pas besoin d'être déclarées avec un type spécifique. Elles sont créées simplement par une initialisation.
- **Toutes les variables sont traitées comme des matrices** : En Matlab, les scalaires sont représentés par des matrices 1x1, les vecteurs par des matrices Nx1 et les matrices par des matrices MxN.

Avantages de Matlab :

- **Rapidité dans l'implémentation et le débogage :** Matlab est connu pour sa facilité d'utilisation et sa rapidité d'implémentation, ce qui en fait un choix populaire pour les prototypages rapides et les simulations.
- **Boîte à outils de traitement d'images riche et puissante :** Matlab offre une boîte à outils spécialisée pour le traitement d'images, qui contient de nombreuses fonctions et algorithmes avancés pour manipuler et analyser des images de manière efficace.

En résumé, Matlab est un environnement de développement puissant et polyvalent, particulièrement adapté à la manipulation de matrices et largement utilisé dans divers domaines scientifiques et techniques.

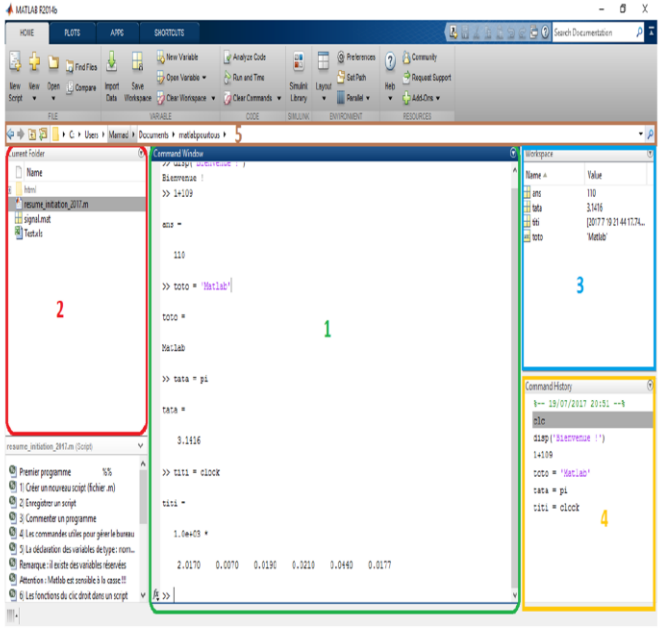

MATLAB R2014b
HOME
PLOTS
APPS
SHORTCUTS
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink Library
Layout
Preferences
Set Path
Parallel
Help
Community
Request Support
Add-Ons
FILE
VARIABLE
CODE
SIMULINK
ENVIRONMENT
RESOURCES
C: ▸ Users ▸ Mamad ▸ Documents ▸ matlabpourtous ▸
5
Current Folder
Name
html
resume_initiation_2017.m
signal.mat
Test.xls
2
resume_initiation_2017.m (Script)
Premier programme %%
1) Créer un nouveau script (fichier .m)
2) Enregistrer un script
3) Commenter un programme
4) Les commandes utiles pour gérer le bureau
5) La déclaration des variables de type : nom...
Remarque : il existe des variables réservées
Attention : Matlab est sensible à la casse !!!
6) Les fonctions du clic droit dans un script
Command Window
Bienvenue !
>> 1+109
ans =
110
>> toto = 'Matlab'
toto =
Matlab
>> tata = pi
tata =
3.1416
>> titi = clock
titi =
1.0e+03 *
2.0170 0.0070 0.0190 0.0210 0.0440 0.0177
>>
1
Workspace
Name
Value
ans 110
tata 3.1416
titi [2017 7 19 21 44 17.74...
toto 'Matlab'
3
Command History
%-- 19/07/2017 20:51 --%
clc
disp('Bienvenue !')
1+109
toto = 'Matlab'
tata = pi
titi = clock
4

La fenêtre principale : C'est la fenêtre principale de MATLAB où vous pouvez voir la barre de menu, la barre d'outils, et la fenêtre de commande. C'est également l'endroit où s'affichent les résultats des commandes que vous exécutez.

L'éditeur de scripts : Cette fenêtre vous permet de créer, modifier et exécuter des scripts MATLAB. Vous pouvez y écrire des lignes de code, les sauvegarder et les exécuter.

La fenêtre de l'Explorateur : Elle vous permet de naviguer dans les fichiers et dossiers de votre système de fichiers, de gérer vos fichiers MATLAB et d'accéder rapidement aux fonctions et variables définies.

Les panneaux latéraux : MATLAB comprend plusieurs panneaux latéraux, tels que le panneau "Variables", qui affiche les variables actuellement définies dans votre espace de travail, et le panneau "Historique des commandes", qui montre les commandes précédemment exécutées.



La barre d'outils : Elle contient des icônes pour les opérations courantes telles que l'exécution de scripts, l'ouverture de fichiers, le débogage, etc.

La barre de menu : Elle offre un accès rapide à toutes les fonctionnalités de MATLAB, organisées par des menus déroulants tels que "Fichier", "Édition", "Affichage", "Outils", etc.

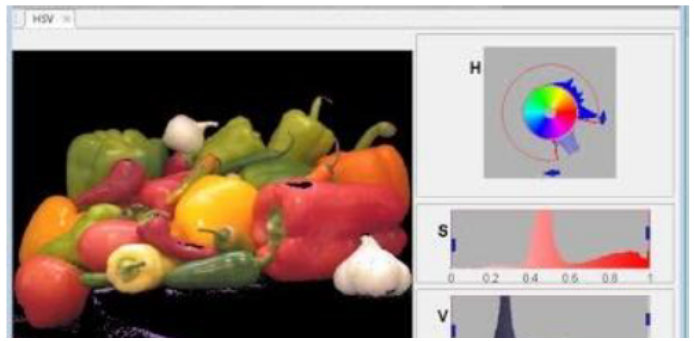
Les fenêtres de graphiques : Lorsque vous tracez des graphiques ou des diagrammes, MATLAB ouvre une fenêtre de graphique séparée pour afficher les résultats. Ces fenêtres offrent des fonctionnalités d'interaction telles que le zoom, la rotation, et l'impression des graphiques.

L'aide en ligne : MATLAB propose une aide contextuelle détaillée pour chaque fonction et commande. Vous pouvez accéder à cette aide en tapant help suivi du nom de la fonction ou en utilisant la fonction doc.

Matlab: Image Processing Toolbox

- Image Processing Toolbox™ propose un ensemble complet d'algorithmes standard de référence et d'applications pour le traitement d'images, l'analyse, la visualisation et le développement d'algorithmes.

- Les opérations que vous pouvez effectuer sont la segmentation des images, l'amélioration des images, la réduction du bruit, les transformations géométriques, le recalage des images et le traitement des images en 3D.



Travaux Pratiques d'Initiation au Traitement d'Images avec MATLAB

Objectif :

Ce TP vise à initier les étudiants au traitement d'images en utilisant MATLAB. Les participants apprendront les bases du chargement, de la manipulation et de l'analyse d'images à l'aide de fonctions et d'outils disponibles dans MATLAB

Exercice 1 : Chargement et Affichage d'une Image

```
% Chargement et affichage d'une image en niveaux de gris
image_gray = imread('cameraman.tif');
info_gray = imfinfo('cameraman.tif');
figure, imshow(image_gray)
title('Image en niveaux de gris')
disp('Informations sur l''image en niveaux de gris :');
disp(info_gray);
```


```
>> Examples
Informations sur l'image en niveaux de gris :
                    Filename: 'C:\Program Files\Polyspace\R2020a\toolbox\images\imdata\cameraman.tif'
                 FileModDate: '04-déc.-2000 18:57:54'
                    FileSize: 65240
                      Format: 'tif'
                 FormatVersion: []
                       Width: 256
                      Height: 256
                    BitDepth: 8
                   ColorType: 'grayscale'
             FormatSignature: [77 77 0 42]
                   ByteOrder: 'big-endian'
                NewSubFileType: 0
                 BitsPerSample: 8
                   Compression: 'PackBits'
    PhotometricInterpretation: 'BlackIsZero'
                 StripOffsets: [8 8262 16426 24578 32492 40499 48599 56637]
              SamplesPerPixel: 1
                 RowsPerStrip: 32
```

imread('cameraman.tif') :

Cette fonction charge une image à partir d'un fichier spécifié ('cameraman.tif' dans ce cas) et stocke son contenu dans la variable **image_gray**.

imfinfo('cameraman.tif') :

Cette fonction extrait les informations détaillées sur l'image spécifiée ('cameraman.tif') et les stocke dans la variable **info_gray.**

figure :

Cette fonction crée une nouvelle figure graphique pour afficher l'image. Toutes les commandes d'affichage ultérieures seront appliquées à cette figure jusqu'à ce qu'une nouvelle figure soit créée ou que le programme se termine.

imshow(image_gray) :

Cette fonction affiche l'image contenue dans la variable **image_gray** sur la figure graphique actuellement active.

title('Image en niveaux de gris') :

Cette fonction ajoute un titre à l'image affichée, indiquant qu'il s'agit d'une image en niveaux de gris.

disp('Informations sur l''image en niveaux de gris :'); :

Cette fonction affiche le texte spécifié dans la fenêtre de commande MATLAB. Dans ce cas, elle affiche un titre pour les informations détaillées sur l'image en niveaux de gris.

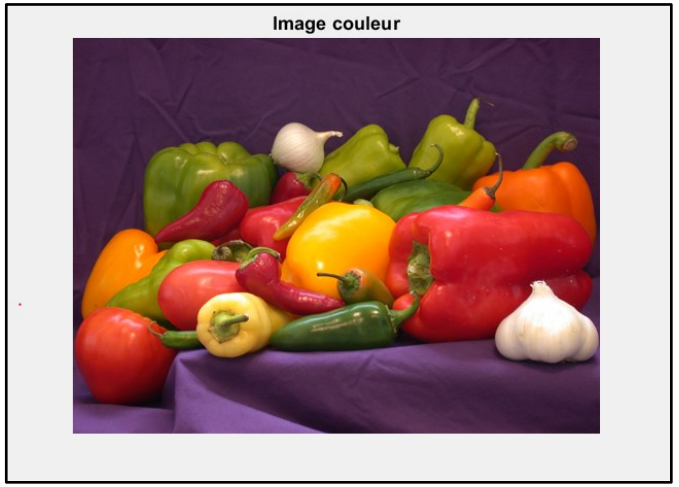
disp(info_gray); :

Cette fonction affiche les informations détaillées extraites de l'image en niveaux de gris, stockées dans la variable info_gray, dans la fenêtre de commande MATLAB.

```
% Chargement et affichage d'une image en niveaux de gris
image_gray = imread('cameraman.tif');
info_gray = imfinfo('cameraman.tif');
figure, imshow(image_gray)
title('Image en niveaux de gris')
disp('Informations sur l''image en niveaux de gris :');
disp(info_gray);
```

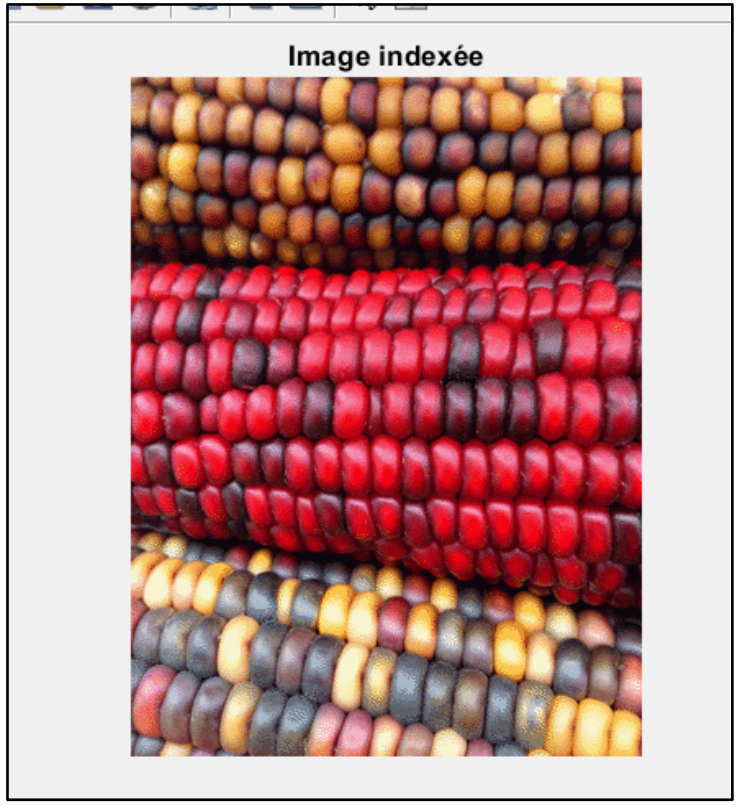
Exercice 1 : Chargement et Affichage d'une Image

```
% Chargement et affichage d'une image couleur
image_color = imread('peppers.png');
info_color = imfinfo('peppers.png');
figure, imshow(image_color)
title('Image couleur')
disp('Informations sur l''image couleur :');
disp(info_color);
```



Exercice 1 : Chargement et Affichage d'une Image

```
% Chargement et affichage d'une image indexée
[image_indexed, map] = imread('corn.tif', 1);
info_indexed = imfinfo('corn.tif');
figure, imshow(image_indexed, map)
title('Image indexée')
disp('Informations sur l''image indexée :');
disp(info_indexed);
```



[image_indexed, map] = imread('corn.tif', 1); :
Cette fonction charge une image indexée à partir d'un fichier spécifié ('corn.tif' dans ce cas) et stocke son contenu dans la variable image_indexed. La carte de couleur associée est stockée dans la variable map.

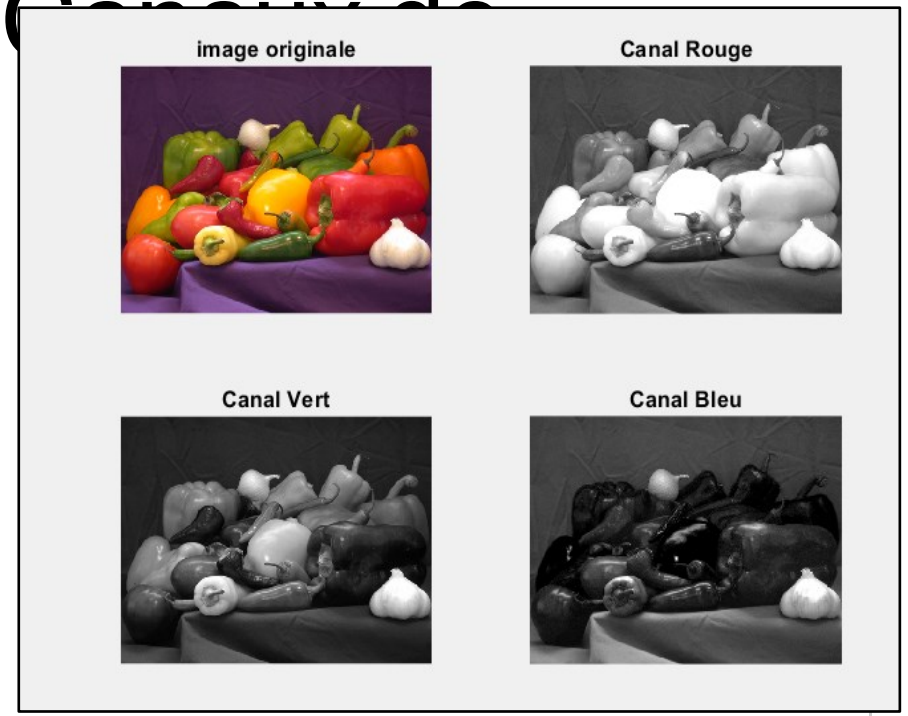
```
% Chargement et affichage d'une image indexée
[image_indexed, map] = imread('corn.tif', 1);
info_indexed = imfinfo('corn.tif');
figure, imshow(image_indexed, map)
title('Image indexée')
disp('Informations sur l''image indexée :');
disp(info_indexed);
```

Dans la fonction imread('corn.tif', 1), l'argument 1 spécifie le numéro de l'image à extraire dans le cas où le fichier 'corn.tif' contient plusieurs images.

Lorsqu'un fichier TIFF contient plusieurs images, chacune peut être associée à un index numérique. Cet index commence généralement à 1 pour la première image et s'incrémente d'une unité pour chaque image subséquente dans le fichier TIFF.

Ainsi, en spécifiant 1 comme deuxième argument de la fonction imread, nous demandons à MATLAB de charger la première image du fichier 'corn.tif'. Si le fichier 'corn.tif' contient uniquement une seule image, spécifier 1 n'affecte pas la sortie de la fonction imread, mais cela reste une convention pour indiquer explicitement l'index de l'image à charger.

Exercice 2 : Manipulation des Canaux de Couleur




```
% Extraire les canaux de l'image couleur
Red = image_color(:,:,1);
Green= image_color(:,:,2);
Blue= image_color(:,:,3);
figure, subplot(2,2,1), imshow(image_color), title('image originale')
subplot(2,2,2), imshow(Red),title('Canal Rouge')
subplot(2,2,3), imshow(Green),title('Canal Vert')
subplot(2,2,4), imshow(Blue),title('Canal Bleu')
```

Explications :

Red = J(:,:,1);, Green= J(:,:,2);, Blue= J(:,:,3); : Ces lignes extraient les canaux Rouge, Vert et Bleu de l'image couleur J. En MATLAB, les images en couleur sont stockées sous forme de matrices en trois dimensions, où la troisième dimension correspond aux canaux de couleur (Rouge, Vert et Bleu).

figure, subplot(2,2,1), imshow(J), title('image originale') : Cette ligne crée une nouvelle figure et divise la fenêtre graphique en une grille 2x2. Elle affiche l'image originale J dans la première sous-fenêtre avec le titre "image originale".

subplot(2,2,2), imshow(Red),title('Canal Rouge') : Cette ligne sélectionne la deuxième sous-fenêtre de la grille et affiche le canal Rouge extrait précédemment Red. Le titre de la sous-fenêtre est "Canal Rouge".

subplot(2,2,3), imshow(Green),title('Canal Vert') : Cette ligne sélectionne la troisième sous-fenêtre de la grille et affiche le canal Vert extrait Green. Le titre de la sous-fenêtre est "Canal Vert".

subplot(2,2,4), imshow(Blue),title('Canal Bleu') : Cette ligne sélectionne la quatrième sous-fenêtre de la grille et affiche le canal Bleu extrait Blue. Le titre de la sous-fenêtre est "Canal Bleu".

Exercice 3 : Conversion d'Images

```
% Conversion en niveaux de gris avec rgb2gray
gray_image_rgb2gray = rgb2gray(image_color);
figure, imshow(gray_image_rgb2gray)
title('Image en niveaux de gris (rgb2gray)')

% Conversion en niveaux de gris avec formule de pondération
gray_image_manual = 0.2989 * Red + 0.5870 * Green + 0.1140 * Blue;
figure, imshow(gray_image_manual)
title('Image en niveaux de gris (formule de pondération)')
```

gray_image_rgb2gray = rgb2gray(image_color); :

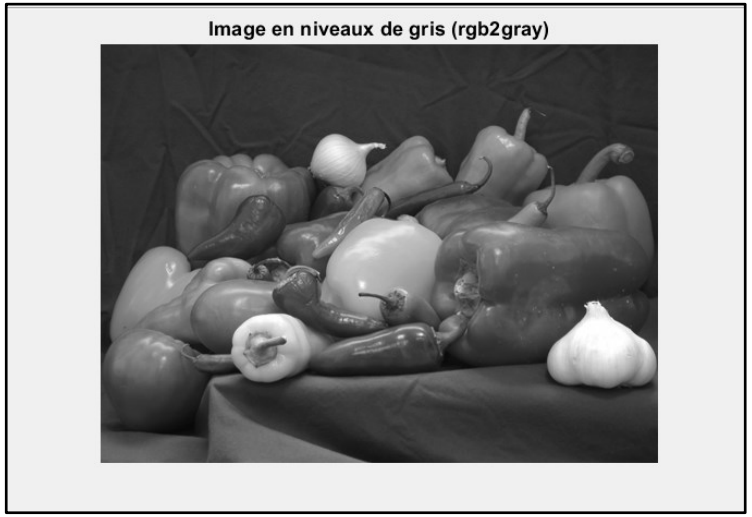
La fonction rgb2gray convertit une image couleur en niveaux de gris en prenant en compte la luminance des couleurs. Elle prend en entrée une image couleur et renvoie une version équivalente en niveaux de gris.
image_color est l'image couleur d'origine.
gray_image_rgb2gray est l'image résultante convertie en niveaux de gris.

figure, imshow(gray_image_rgb2gray) :

Cette ligne crée une nouvelle figure et affiche l'image convertie en niveaux de gris à l'intérieur de cette figure.
imshow est une fonction utilisée pour afficher une image dans une figure.
gray_image_rgb2gray est l'image en niveaux de gris à afficher.

title('Image en niveaux de gris (rgb2gray)') :

Cette ligne ajoute un titre à la figure affichant l'image en niveaux de gris convertie à l'aide de rgb2gray.



$$\text{Luminance} = 0.2989 \times \text{Rouge} + 0.5870 \times \text{Vert} + 0.1140 \times \text{Bleu}$$

gray_image_manual = 0.2989 * Red + 0.5870 * Green + 0.1140 * Blue; :

Cette ligne réalise une conversion manuelle d'une image couleur en niveaux de gris en appliquant une formule de pondération. Les poids utilisés sont basés sur la sensibilité de l'œil humain aux différentes couleurs.
Red, Green et Blue sont les canaux de couleur extraits de l'image couleur d'origine.
gray_image_manual est l'image en niveaux de gris résultante après application de la formule de pondération.

figure, imshow(gray_image_manual) :

Cette ligne crée une nouvelle figure et affiche l'image convertie en niveaux de gris manuellement à l'intérieur de cette figure.

title('Image en niveaux de gris (formule de pondération)') :

Cette ligne ajoute un titre à la figure affichant l'image en niveaux de gris convertie à l'aide de la formule de pondération.

Explication de la formule :

- ✓ Lorsque vous regardez une image couleur, comme une photo, chaque pixel de cette image contient trois informations : combien de **rouge**, combien de **vert** et combien de **bleu** il doit afficher pour créer la couleur que vous voyez. Mais pour créer une image en niveaux de gris, vous n'avez besoin que d'une seule information par pixel : **à quel point ce pixel est lumineux.**
- ✓ Donc, pour convertir une image couleur en niveaux de gris, vous devez combiner ces trois informations **(rouge, vert, bleu)** en une seule information (luminance), qui représente à quel point le pixel est lumineux.
- ✓ Maintenant, les humains voient le vert plus facilement que le rouge, et le rouge plus facilement que le bleu. Donc, lors de la conversion, nous accordons plus de poids au vert, puis un peu moins au rouge, et encore moins au bleu.
- ✓ La formule que nous utilisons ressemble à ceci : $\text{Luminance} = 0.2989 \times \text{Rouge} + 0.5870 \times \text{Vert} + 0.1140 \times \text{Bleu}$
- ✓ Ce que cette formule fait, c'est prendre chaque valeur de rouge, vert et bleu, la multiplier par un poids spécifique (0.2989 pour le rouge, 0.5870 pour le vert, 0.1140 pour le bleu), puis ajouter toutes ces valeurs ensemble pour obtenir une seule valeur qui représente à quel point ce pixel est lumineux. Et voilà ! Maintenant, vous avez une image en niveaux de gris, où chaque pixel est simplement une nuance de gris qui représente sa luminosité.