

IMAGERIE SATELLITAIRE

---

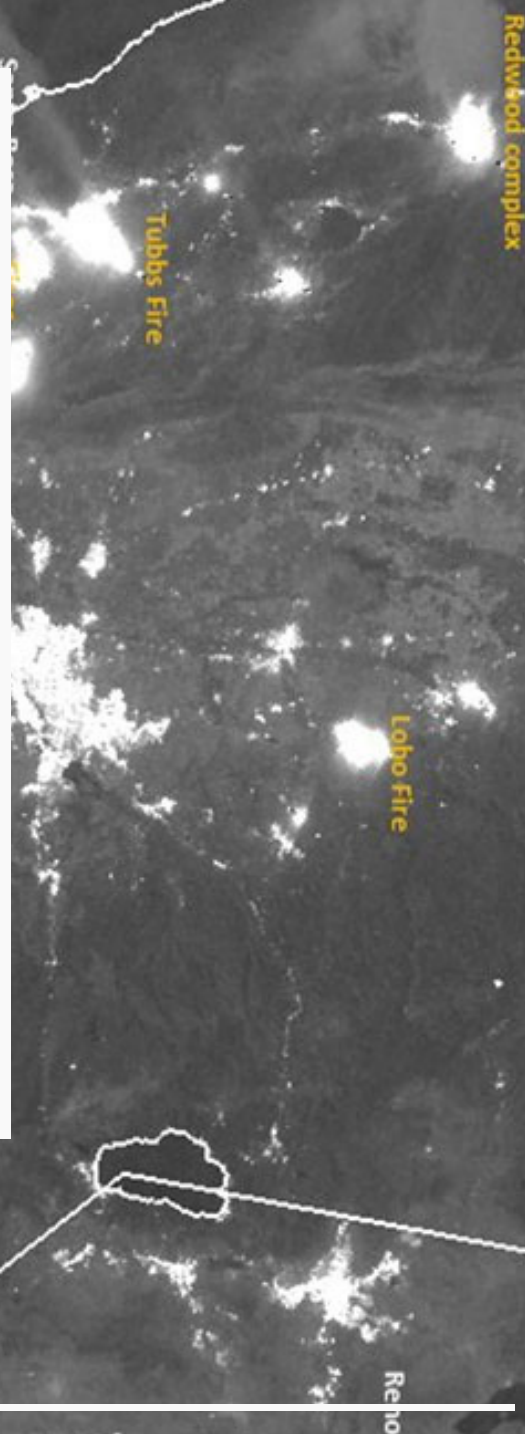
# RAPPORT TP-01-ISAT

**PRÉSENTÉ PAR**  
HANIFI FATIMA EL-BATOUL  
**GROUPE : 02 IAA**

**MME.BEKKOUCHE**

---

**NOVEMBRE 2021**



# AVANT-PROPOS

Ce manuel est le rapport des travaux pratiques de télédétection du cours imagerie satellitaire dispensé sur le département d'informatique ,Université d'USTO, Oran.

Dans ce rapport je vais expliquer en détails les traitements appliquer sur les images satellitaires prise par un satellite contient plusieurs capteurs dans notre cas nous travaillerons avec 4 canaux TM1, TM2, TM3, TM4.

L'environnement de développement principalement utilisé dans ce TP est Jupyter Notebook version 6.1.4, utilisant le langage de programmation PYTHON3.



**IMAGERIE  
SATELLITAIRE**

# INTRODUCTION

Dans une image satellitaire, l'information sur les couleurs est décomposée en différents canaux ou bandes spectrales. Chaque bande est une image en niveaux de gris, composée de pixels ayant chacun une valeur de réflectance pour un intervalle de longueur d'ondes donné.

Une image satellitaire, selon les caractéristiques du capteur embarqué sur le satellite, peut contenir en plus des trois bandes du visible (RVB ou RGB) quelques bandes supplémentaires (par exemple infrarouge, proche infrarouge), et jusqu'à des centaines de bandes. Ces bandes vont couvrir des intervalles plus ou moins large du spectre électromagnétique. On parle ainsi d'images multi-spectrales (quelques bandes) ou d'images hyper-spectrales (des dizaines à des centaines de bandes).

Le choix des images satellitaires utilisés dans ce rapport, selon l'énoncé sont de taille 800 x 400 pixels prise par le satellite LANDSAT 5 TM (Thematic Mapper) en 2003 de la région d'Oran nord-ouest. Sachant que ce satellite contient plusieurs capteurs dans notre cas nous travaillerons avec 4 canaux 1, 2, 3 et 4.

Dans ce rapport de TP qui s'intitule initialisation au traitement d'image satellitaires, nous allons faire quelque traitements de base (charger et afficher une images, faire des compositions colorées etc...) sur celles-ci.

# EXPLICATION DE CODE

Nous allons voir comment charger une image et faire quelques opérations de base sur celles-ci

## Etape1 : ouverture des images satellitaires

Importez tout d'abord les librairies Python, comme suit :

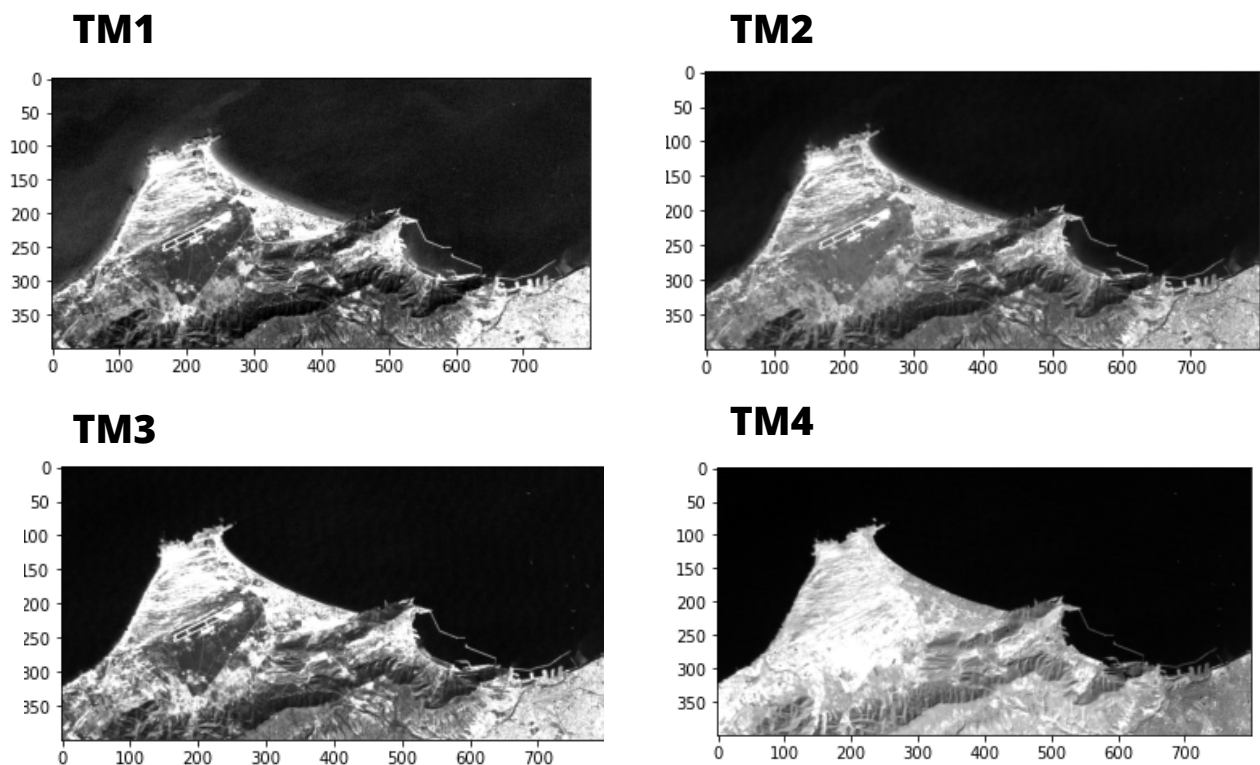
```
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
import cv2
from skimage import exposure
```

Ensuite il suffit de lire les image avec la fonction imread() de matplotlib :

```
TM1 = plt.imread('TM1.BMP') # Lire l'image TM1.BMP
TM2 = plt.imread('TM2.BMP') # Lire l'image TM2.BMP
TM3 = plt.imread('TM3.BMP') # Lire l'image TM3.BMP
TM4 = plt.imread('TM4.BMP') # Lire l'image TM4.BMP
```

Afficher les image TM1, TM2, TM3, TM4, tout simplement avec la fonction imshow() :

```
#affichage des images TM1, TM2, TM3, TM4
plt.imshow(TM1)
plt.show()
plt.imshow(TM2)
plt.show()
plt.imshow(TM3)
plt.show()
plt.imshow(TM4)
plt.show()
```



**FIGURE 1 IMAGE SATELLITAIRE DES 4 CANAUX TM1, TM2, TM3, TM4**

## Analyse :

ces image, selon l'énoncé sont des image d'un certain niveau de couleur.

entre: 0 et 255.

0 : noir.

255 : blanc

si un élément est plus proche de 255, cela veut dire que cet élément est apparent et si un élément est plus proche de 0, cela veut dire que cet élément n'est pas apparent, et autre éléments partiellement apparents entre 0 et 255.

selon mes analyses, les images donnés, contient tout ces élément, telle que:

### TM1:

- Route et endroit habitable (bâtiments ...) : apparent avec une intensité similaire
- végétation : aucun
- Montagne terrestre :aucun
- Mer : oui
- roches : oui

**TM2:**

- Route et endroit habitable (bâtiments ...) : apparent avec une intensité similaire
- végétation : partiellement
- Montagne terrestre : partiellement
- Mer : non
- roches : oui

**TM3:**

- Route et endroit habitable (bâtiments ...) : apparent avec une intensité similaire
- végétation : oui
- Montagne terrestre : non
- Mer : non
- Roches : oui

**TM4:**

- Route: oui
- végétation : oui
- Montagne terrestre : partiellement
- Mer : non
- Endroit habitable (bâtiments ...) : non
- Roches : oui



## Etape 2 : affichage d'une image en couleur

Cette partie de code s'est surtout basé sur l'utilisation des compositions colorées Rouge (R), Vert (V), Bleu (B).

### 2-1 Le principe de la composition colorée:

Sur une image classique, on reconstitue à l'écran ce que notre œil voit en affectant un filtre coloré correspondant à la "vraie" couleur de chaque bande, (c'est le principe de la synthèse additive : toutes les couleurs du spectre visible additionnées, ou les 3 couleurs primaire additionnées donnent de la lumière blanche). On parle alors de deux type de **composition colorée vraies couleurs**, le principe est le même pour les images satellitaires.

#### 2-1-1 composition colorée vraies couleurs:

si on souhaite afficher une images optique telle que notre œil verrait la Terre depuis le ciel, on affectera aux trois bandes du visible bleu, du vert et du rouge leurs vraies couleurs. On a un autre type de les compositions colorées c'est compositions des fausses couleurs.

Mais on a un autre type de de composition colorée c'est:

#### 2-1-2 composition colorée fausses couleurs:

Si on affecte à la bande du proche infrarouge un filtre rouge, à la bande du rouge un filtre verte et à la bande du vert un filtre bleu. On parle alors de composition colorée "fausses couleurs".

## A-COMPOSITION TM3/TM2/TM1 "COMPOSITION COLORÉE VRAIES COULEURS" :

Composition colorée 3-2-1 (couleurs naturelles) , Pour pouvoir procéder au contrôle de la réalité du terrain, la composition 321 qui correspond aux couleurs naturelles, a été utilisée comme le montre l'image resultante ci-après.

Pour réaliser cette composition, on crée une matrice de zéros d'une image initial, en suite on lit les images utilisant `plt.imread('TM3.BMP')` # lire TM3, et la garder dans une variable `IMG3....`(Cette partie de code est bien expliquer dans les commentaires.)

```
sizeX=400 #Largeur
sizeY=800 #Longeur

imgRGB=np.zeros([sizeX, sizeY,3]) #creation d'une matrice d'une image initial

IMG3=plt.imread('TM3.BMP') #lecture de TM3.BMP au variable IMG3
IMG2=plt.imread('TM2.BMP') #lecture de TM2.BMP au variable IMG2
IMG1=plt.imread('TM1.BMP') #lecture de TM1.BMP au variable IMG1

#assignation des image au different niveau de couleur RGB du imgRGB
imgRGB[:, :,0]=IMG3[:, :,0]/255 #niveau rouge
imgRGB[:, :,1]=IMG2[:, :,1]/255 #niveau vert
imgRGB[:, :,2]=IMG1[:, :,2]/255 #niveau bleu

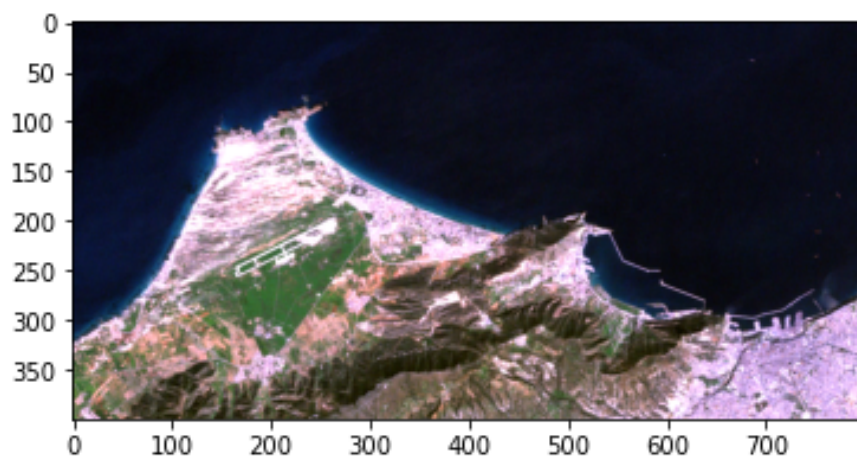
#on à du dans les 3 dernieres instruction diviser sur 255 car les matrice
#des images IMG1,IMG2,IMG3 sont du format entier, et imgRGB est flottante
#dans ce cas la conversion de IMG1,IMG2,IMG3 est obligatoire ce qui est fait
#par division sur 255

plt.imsave("ImgCouleur1.bmp",imgRGB,cmap="gray")
#sauvegarder l'image avec le titre "ImgCouleur.bmp" est utilisé
#pour avoir l'image couleur resultante de cette cellule en format bmp

plt.imshow(imgRGB)
plt.show()
```



Sur l'illustration ci-après, sur la composition colorée vraie couleur la ville apparaît avec des couleurs blanches et grises, l'eau dans différentes teintes de bleu, et la végétation en vert foncé : on a recréé ce que verrait notre œil.



**FIGURE 2 COMPOSITION TM3/TM2/TM1  
"COMPOSITION COLORÉE VRAIES COULEURS"**

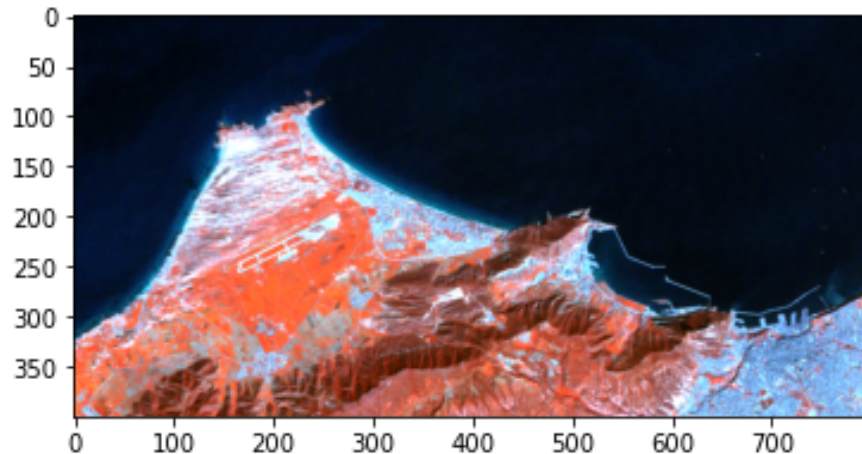
## **B-COMPOSITION TM4/TM2/TM1 "COMPOSITION COLORÉE FAUSSE COULEURS"**

```
IMG3=plt.imread('TM4.BMP')
IMG2=plt.imread('TM2.BMP')
IMG1=plt.imread('TM1.BMP')

imgRGB[:, :, 0]=IMG3[:, :, 0]/255
imgRGB[:, :, 1]=IMG2[:, :, 1]/255
imgRGB[:, :, 2]=IMG1[:, :, 2]/255

plt.imsave("ImgCouleurB.bmp",imgRGB,cmap="gray")

plt.imshow(imgRGB)
plt.show()
```



**FIGURE 3 COMPOSITION TM4/TM2/TM1  
"COMPOSITION COLORÉE FAUSSE COULEURS"**

## 2-2) Est-il possible de faire d'autres compositions ?

Oui possible de faire autre composition, car selon les remarques du résultat du question précédente, si on superpose chaque images dans des niveau de couleurs: on peux trouver la meilleur image possible, on peut faire plusieurs composions par rapport au 4 canaux TM1, TM2, TM3, TM4, tout dépend de ce qu'on cherche,

si on cherche la mer par exemple, on fait une composition qui sort vraiment la mer et si on veut classer tout on cherche une composition qui sort vraiment tout.

Dans notre cas, on peut classer les niveaux comme suit:

### **Le niveau rouge:**

TM2-TM4 car seul l'élément rouge c'est les montagnes terrestre (marron au lieu rouge mais il se compose du rouge) et exclure également TM2 car on veut pas la végétation rouge)

### **Le niveau vert:**

TM2 et TM3 car il montrent la végétation partiellement bien.

### **Le niveau bleu:**

TM1 car il montre bien la mer.

donc les choix possibles (1\*2\*1) a l'ordre RGB on peut faire:

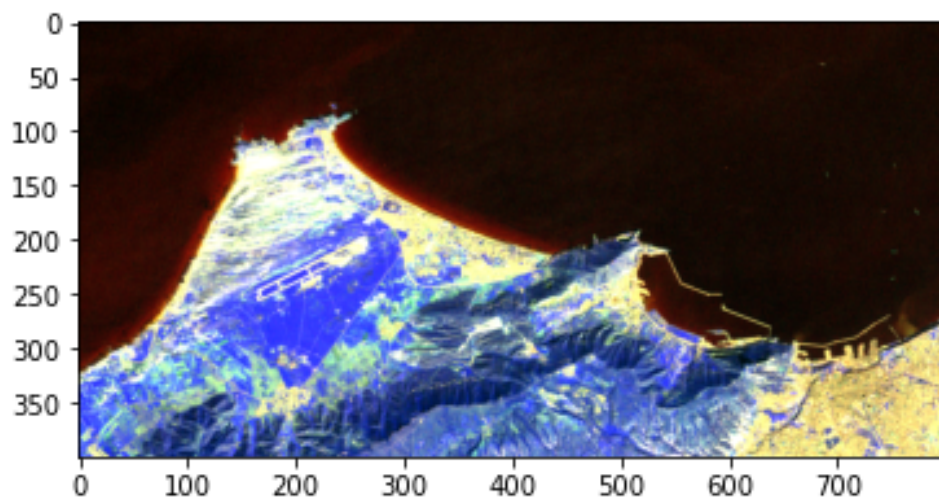
TM4-TM3-TM1

TM4-TM2-TM1

## 2-3 Quelques compositions supplémentaire:

### 2.3.1 Composition TM1 TM3 TM4:

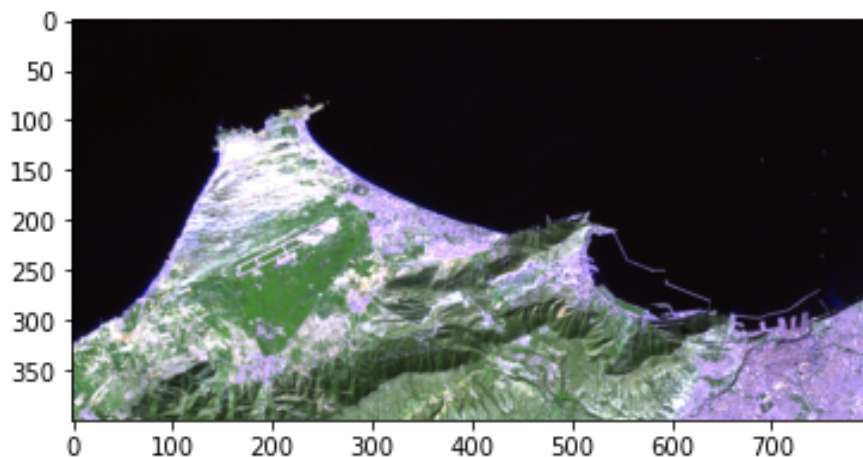
Une image en fausse couleur composée de trois bandes a été obtenue à partir de la superposition des trois canaux TM1, TM3, TM4 auxquels des pseudo-couleurs ont été attribués et qui sont respectivement : rouge, vert, bleu. Elles sont suffisantes pour l'étude de l'ensablement.



**FIGURE 4 COMPOSITION TM1 TM3 TM4  
"COMPOSITION COLORÉE FAUSSE COULEURS"**

### 2.3.2 Composition TM7 TM5 TM3:

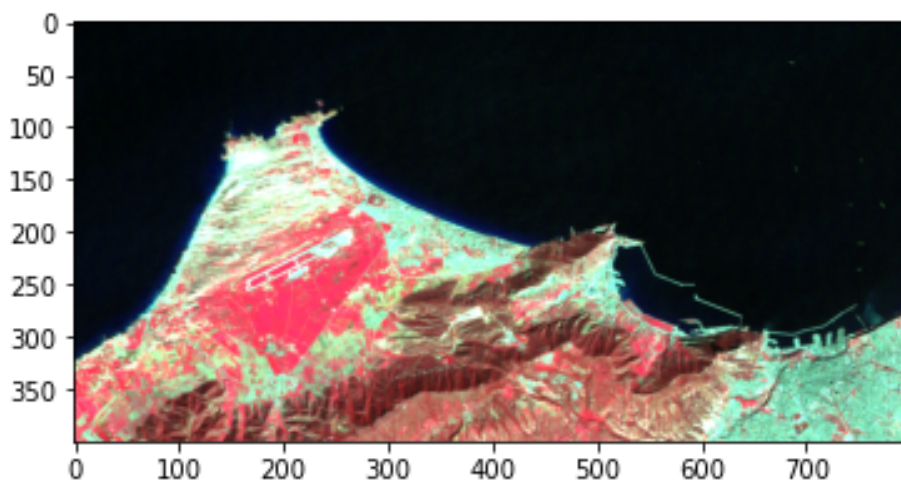
Compte tenu des propriétés des calcaires qui généralement se présentent sous une coloration claire, une faible porosité (donc faible humidité) dans le cas où l'on n'observe pas de dissolution secondaire, l'utilisation de la composition colorée 753 a permis de mettre en évidence la limite des dépôts de calcaire tel que l'on peut l'observer sur l'image qui suit.



**FIGURE 5 COMPOSITION TM7 TM5 TM3**

### 2.3.3 Composition TM4 TM3 TM2:

La combinaison 432 met en évidence la couverture végétale qui suppose donc l'existence de zones humides, en rouge sang sur l'image. Par opposition, les zones dépourvues de couverture végétale apparaissent en clair.



**FIGURE 6 COMPOSITION TM4 TM3 TM2**

## Conclusion:

La réalisation d'une composition colorée consiste à attribuer à chaque couleur primaire (rouge, vert et bleu) trois bandes spectrales d'un capteur satellitaire. Par synthèse additive, il est possible de reconstituer toutes les couleurs. Cela permet de faciliter l'interprétation des images satellitaires et de mettre en évidence des phénomènes environnementaux.



## Etape 3 : affichage l'histogramme des images satellitaires en nuances de gris

### Définition:

Un histogramme d'image n'est ni plus ni moins qu'un graphique qui affiche :

- En abscisse (Val. dans le graphe ci-dessous) les différentes valeurs de canaux/pixel
- En ordonnée (Nb Pixel) le nombre de canaux/pixel qui possèdent cette valeur

### 3-1 Affichage l'histogramme aux niveaux de gris pour les trois images TM1, TM2 et TM3:

Une image en niveau de gris ne possède qu'une seule courbe (celle des nuances de gris):

### Histogrammes avec scikit-image:

Avec scikit-image nous utiliserons simplement la fonction histogram pour tracer ces graphes.

```
def imageHist(image):
    _, axis = plt.subplots(ncols=2, figsize=(18, 3))
    if (image.ndim == 2): #Si on a 3 dimensions
        # Grayscale Image
        axis[0].imshow(image, cmap=plt.get_cmap('gray'))
        axis[1].set_title('Histogram') #titre pour l'histogramme
        axis[0].set_title('Grayscale Image') #titre pour l'image
        hist = exposure.histogram(image) #Return l'histogramme de l'image.
        axis[1].plot(hist[0])
    else:
        # Color image
        axis[0].imshow(image, cmap='gray')
        axis[1].set_title('Histogram') #titre pour l'histogramme
        axis[0].set_title('Colored Image') #titre pour l'image
        rgbcolors = ['red', 'green', 'blue'] #RGB
        for i, mycolor in enumerate(rgbcolors):
            axis[1].plot(exposure.histogram(image[...,i])[0], color=mycolor)
            #Return l'histogramme de l'image.

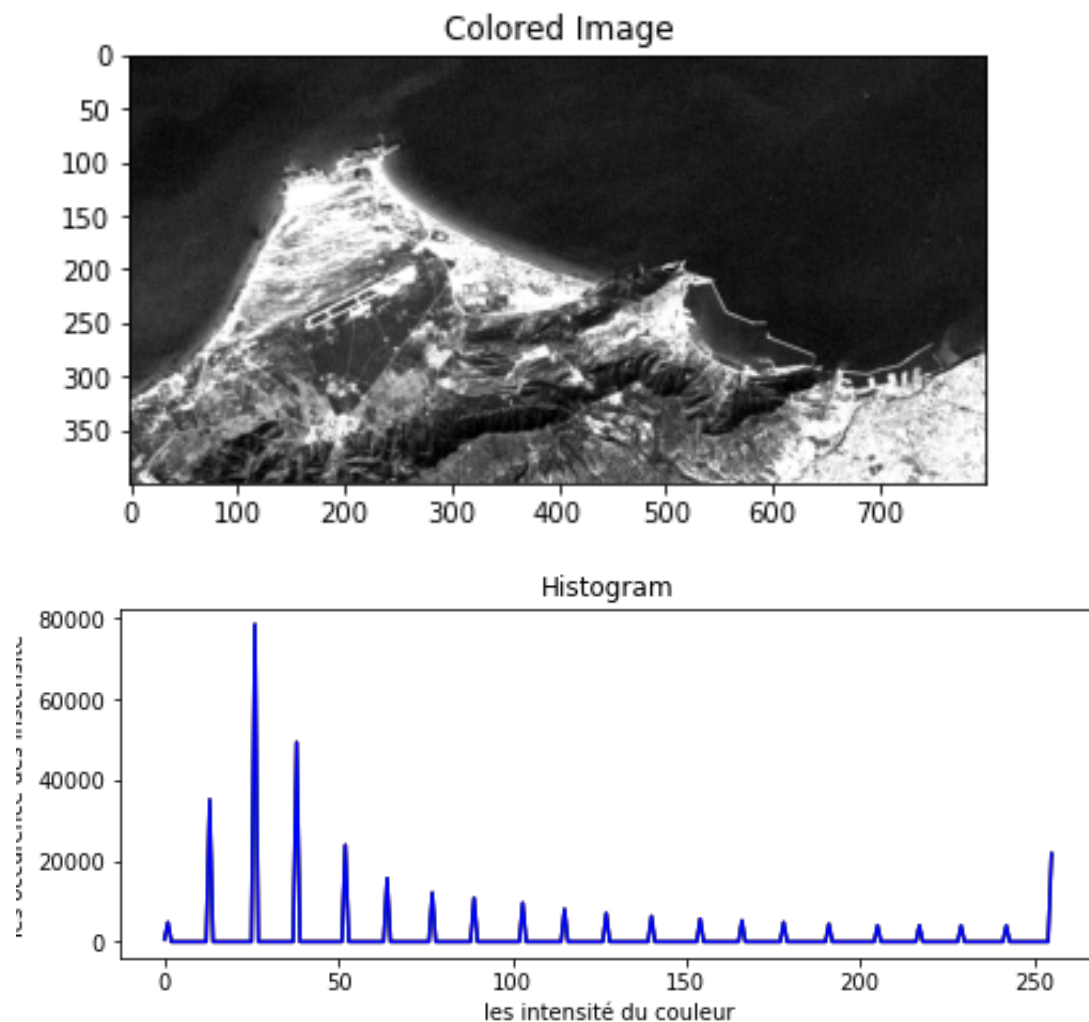
    plt.legend(['Rouge', 'Vert', 'Bleu']) #fonction du legende (les attributs ont
                                         # Le meme ordre avec les 3 plots precedent)
    plt.xlabel('les intensité du couleur')
    plt.ylabel('les occurrence des instensite')
```

**Explication de code:**

Dans le code ci-dessus on regarde en premier lieu le nombre de dimensions de la matrice image. Si on a 3 dimensions c'est que l'on a une image avec couleur sinon on est en niveau de gris. Dans le cas où l'on a une image avec couleur on doit empiler les canaux et donc nous aurons 3 courbes. comme suit :

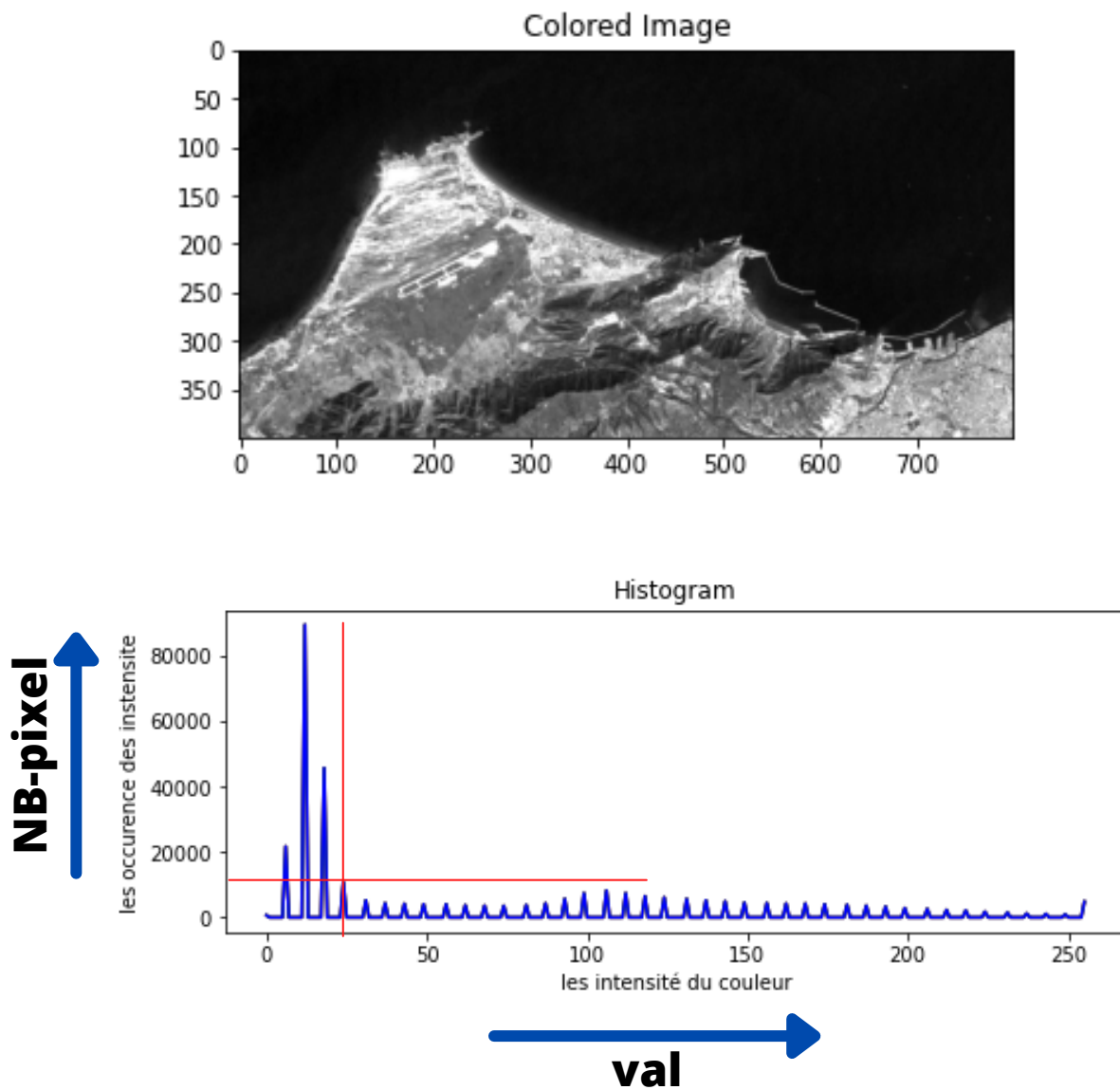
On fait l'appel a la fonction `imageHist()` pour afficher l'image et l'histogramme tout simplement :

```
imageHist(TM1)
```



**FIGURE 7 IMAGE EN COULEUR ET L'HISTOGRAMME DE L'IMAGE TM1.**

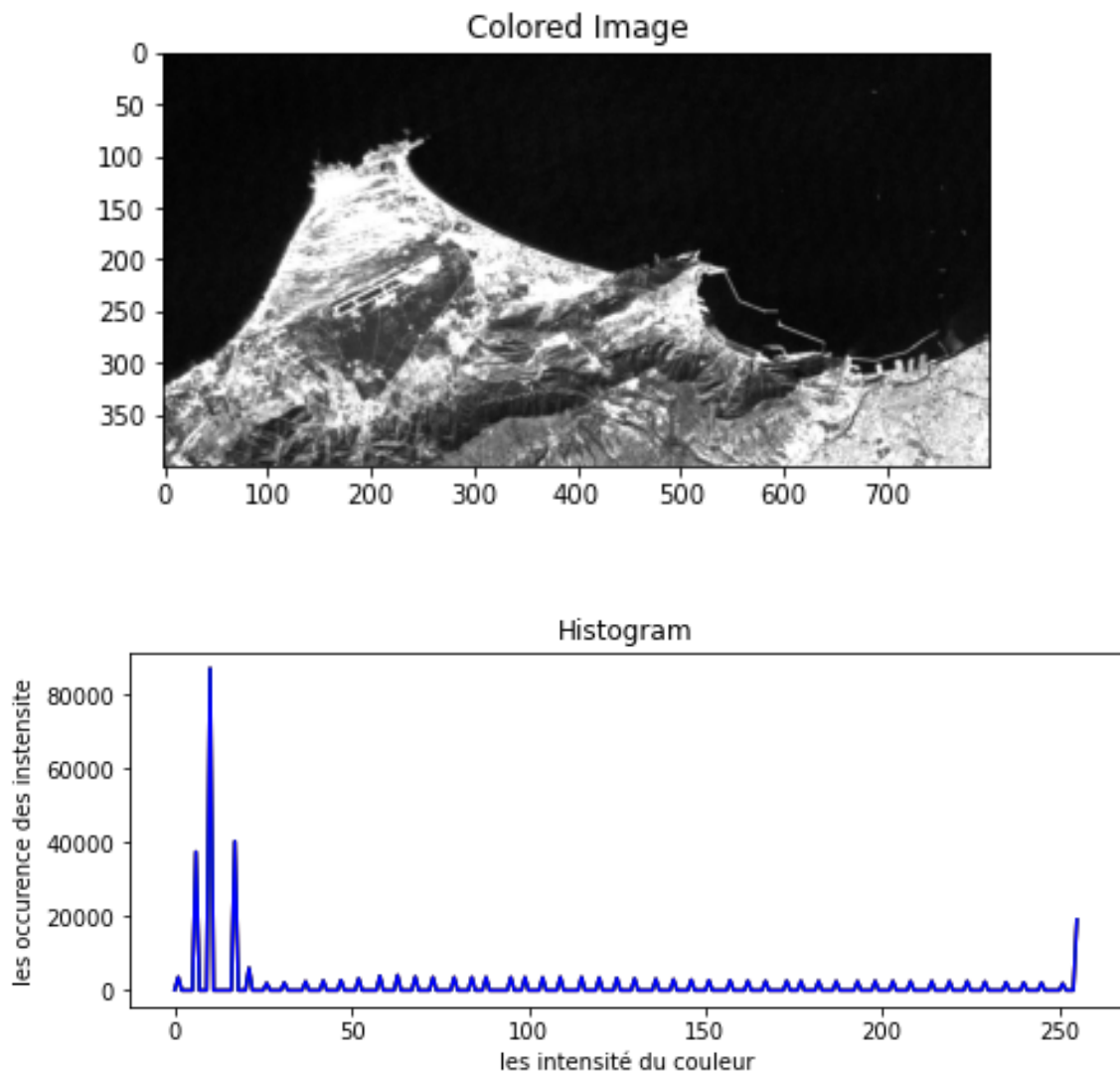
```
imageHist(TM2)
```



**FIGURE 8 IMAGE EN COULEUR ET L'HISTOGRAMME DE L'IMAGE TM2.**

Dans le graphe ci-dessus, on a 1000 fois un pixel de valeur 25.

```
imageHist(TM3)
```



**FIGURE 9 IMAGE EN COULEUR ET L'HISTOGRAMME DE L'IMAGE TM2.**

### Remarque :

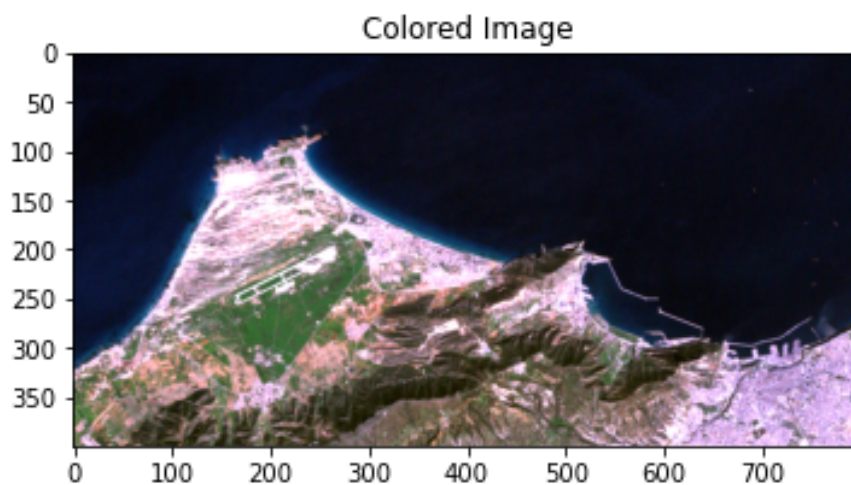
une seule courbe. Cela signifie tout simplement que nous sommes en train de visualiser une image en niveau de gris de TM1, TM2, et TM3.

Donc une image en niveau de gris ne possède qu'une seule courbe (celle des nuances de gris).

### 3-2 Est-il possible de faire pour une image en couleur ?

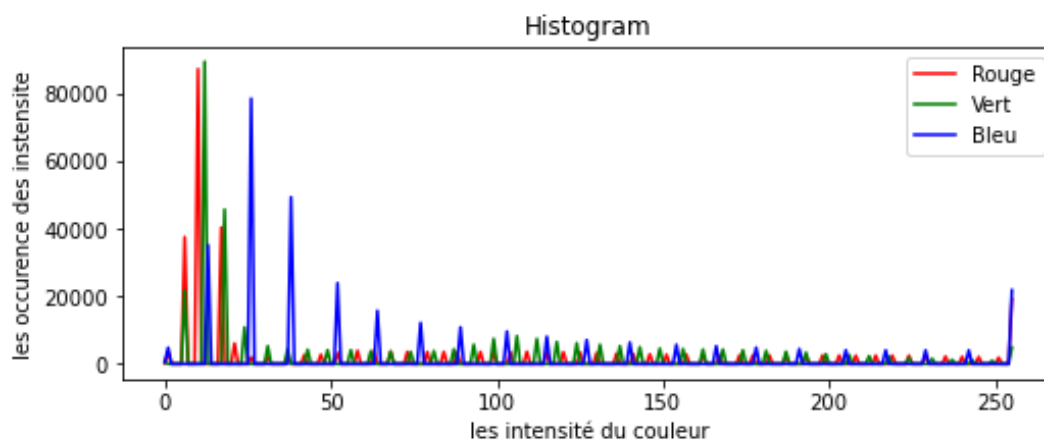
Oui, possible de faire l'histogramme pour une image en couleur. Chaque pixel étant donc un tuple ou plutôt une superposition de canaux de couleurs (Rouge, Vert et Bleu, soit [R, G, B]), nous allons devoir analyser cette répartition des trois couleurs primaires dans notre image.

Nous allons donc voir dans cette partie comment analyser les composantes de couleurs à travers d'histogrammes.



**FIGURE 10 IMAGE EN COULEUR DE LA COMPOSITION TM3 TM2 TM1.**

Si on a une image en couleur il nous faut maintenant avoir les nuances sur les trois canaux (Rouge, Vert et Bleu). On a donc 3 courbes comme suit:



**FIGURE 11 L'HISTOGRAMME DE L'IMAGE EN COULEUR DE LA COMPOSITION TM3 TM2 TM1.**

Voici la partie code correspondante à la lecture de l'image couleur avec `plt.imread()`, déjà sauvegarder (voir partie code précédente) dans `"ImageCouleur1.BMP"`, cette image est le résultat des 3 compositions TM3 TM2 TM1.

```
IMGCLR=plt.imread("ImgCouleur1.bmp")
imageHist(IMGCLR)
```

## Analyse :

Dans l'histogramme on remarque 3 courbes, vert, rouge , puis le bleu , cela veut dire:

- le vert puis le rouge avec un petit écart ==> végétation en vert et les routes qui sont en rouge (difficilement remarquable).

- le bleu, les intensités bleu est moins par rapport au vert et rouge mais pas rare ==> la mer en bleu.

On remarque une bonne clarté et précision..

## Conclusion:

l'histogramme des couleurs est plus efficace que le tâtonnement sur les éléments triviaux : la mer est bleu, la végétation est verte , les routes en rouges etc..).



## Références:

1- Saley Mahaman.B , Wade.S, *Extraction par télédétection et analyse statistique du réseau de fractures, en milieu de socle, dans le département de Sinfra. Journal, 2015,. 13 No.*

2-MOANAZAFY.S, *CARTOGRAPHIE DES STRUCTURES TECTONIQUES ET TRAÇAGE DES APPORTS TERRIGÈNES PAR IMAGERIE SATELLITAIRE. Memoire de fin d'etude, UNIVERSITE D'ANTANANARIVO , 2009.*

3-PÔLE MÉTIER, *Comprendre une image satellitaire.*GeoBretagne. 2016

A satellite image of a storm system, likely a tropical cyclone, with a teal overlay indicating a specific data layer. The image is overlaid with a grid of white crosses. A horizontal white line is positioned below the title.

**IMAGERIE SATELLITAIRE**

**MERCI.**

**NOVEMBRE 2021**