

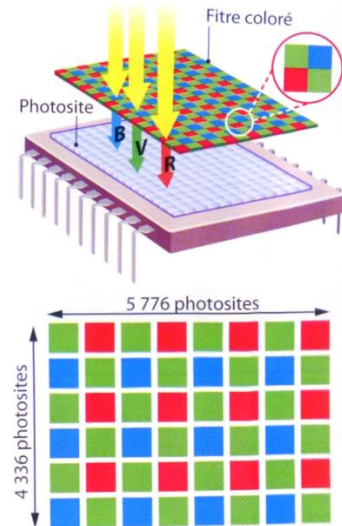
Partie IV/ Photographie numérique

I/ La capture d'une image

Le capteur

Un **capteur** photo transforme l'énergie lumineuse en un signal électrique. Il est composé d'une grille de **photosites**, c'est-à-dire de cellules mesurant l'intensité lumineuse reçue.

Afin de mesurer les valeurs des couleurs entrantes dans l'appareil, chaque photosite est recouvert d'un filtre coloré ne laissant passer que les rayons d'une seule couleur : le **rouge**, le **vert** ou le **bleu**.



Capteur de 25 millions de photosites

Définition d'un capteur : nombre total de ses photosites. Plus la définition est élevée, meilleure sera la qualité de l'image produite.

Couleurs primaires en synthèse additive

Couleurs primaires : les couleurs primaires correspondent au plus petit ensemble de couleurs de base dont la combinaison permet d'obtenir la totalité de toutes les couleurs du spectre visible. Elles sont au nombre de trois.

En les mélangeant en des proportions variables il est possible d'obtenir n'importe quelle teinte de n'importe quelle couleur perceptible par l'œil humain, elles sont donc exploitées pour toute synthèse de couleur (écrans, arts plastiques, éclairages d'ambiance ou artistiques...)

Les couleurs primaires sont au nombre de trois mais ne sont pas les mêmes en **synthèse additive** (rouge, vert, bleu) ou **soustractive** (jaune, cyan, magenta).

Mélanger du rouge, du bleu et du vert en même quantité donne du **blanc**.

En informatique, on tient compte de la quantité de chaque couleur, ainsi, on obtient des teintes grisées, allant du noir (aucune couleur) au blanc (quantité maximale de chaque couleur).

II/ L'image numérique

Caractéristiques d'une image numérique

Une image numérique se présente sous la forme d'un quadrillage dont chaque case est un **pixel** d'une couleur donnée.

Pixel : unité de base composant une image numérique. Il est composé d'au moins trois nombres qui correspondent aux quantités de rouge, bleu et vert. Il peut aussi comporter une autre valeur, correspondant à la transparence de l'image.

Définition de l'image : nombre total de pixels qui la composent. Plus elle est élevée, plus la résolution de l'image est meilleure.

Codage des pixels

Chaque pixel correspond à un triplet de trois nombres qui sont les valeurs du rouge, vert et bleu comprises entre **0** (pas de couleur) et **255** (quantité maximale). Ce code est appelé **code RVB**, il en existe d'autres.

Les **teintes grisées** ont donc la **même valeur pour chaque canal**.

R = 255, V = 0, B = 0
R = 0, V = 255, B = 0
R = 0, V = 0, B = 255
R = 238, V = 160, B = 73
R = 175, V = 18, B = 204
R = 0, V = 0, B = 0

Question : Pourquoi utiliser cette plage de valeurs de 0 à 255 ?

.....

III/ Les formats et métadonnées

Format d'une image

Tout image acquise et stockée sous forme binaire (0 et 1) est dite **numérique**. Son **format** est la manière dont est codée cette suite de 0 et de 1.

Lors de la capture par l'appareil photo, un fichier de format **brut** (*en anglais Raw*) qui contient toutes les données brutes issues du capteur. Elles sont ensuite sauvegardées sous un autre format que l'on peut connaître grâce à l'extension du fichier (.jpeg ; .png ; .bmp ; .gif etc.).

Ces différents formats permettent d'utiliser à bon escient la qualité souhaitée et le poids en octets de l'image que l'on peut diminuer grâce aux **algorithmes de compression** d'images.

FORMAT	QUALITÉ	POIDS
Raw	Maximale	Très volumineux
Tiff	Excellente	Volumineux
Jpeg	Variable	Peu volumineux

Caractéristiques de trois formats d'images

Métadonnées EXIF

Au moment de la création du fichier Raw, de nombreuses données relatives à la prise de vue sont enregistrées. Il s'agit des **métadonnées**, enregistrées dans un fichier au format EXIF. Elles comprennent entre autres la date, l'heure, les paramètres de prise de vue (vitesse, sensibilité etc.), la compression, la géolocalisation de l'image.

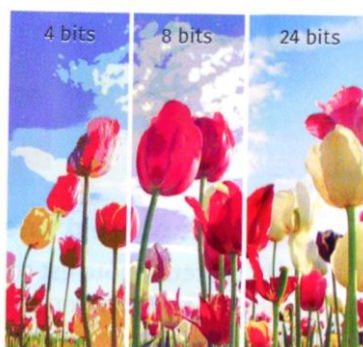
IV/ Transformer une photo par des algorithmes et logiciels

Des transformations multiples

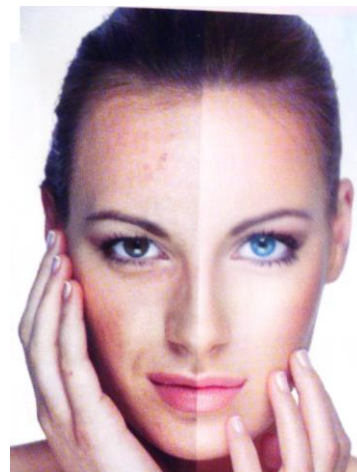
De nombreux algorithmes interviennent au cours de la capture et du traitement de l'image par l'appareil photo numérique. Ils effectuent différentes opérations lors de la prise de vue (exposition au soleil, stabilisation) mais aussi lors du traitement automatisé de l'image (netteté, rendu des couleurs).

De nombreux logiciels, **GIMP** et **Photoshop** pour les plus connus permettent de retoucher et modifier les images. On peut alors très facilement « truquer » une photo.

Faire paraître une photo d'une personne (transformée ou non) nécessite évidemment son accord. Cela est **le droit à l'image et à la dignité**, la loi encadre ce dispositif.



Des qualités différentes d'images



Visage sans et avec retouche photo

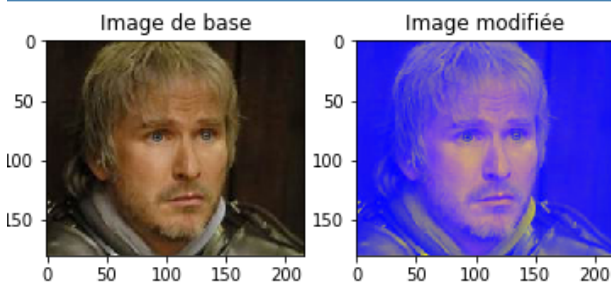
Modifier des photos avec Python

Analyse d'un programme Python inversant le canal bleu d'une image.

```
1 from PIL import Image # Importe la bibliothèque gérant les images
2 import matplotlib.pyplot as plt # Importe la bibliothèque permettant ici d'afficher les images
3 import requests # Importe les bibliothèques permettant d'utiliser une image sur le net
4 from io import BytesIO
5
6 # Affiche l'image de base et celle transformée
7 def showImages(img1,img2) :
8     # Permet de créer une sous_fenêtre, le 3eme arg est la place de l'image
9     plt.subplot(1,2,1)
10    plt.title("Image de base")
11    plt.imshow(img1,cmap = 'gray')
12    plt.subplot(1,2,2)
13    plt.title("Image modifiée")
14    plt.imshow(img2,cmap = 'gray')
15
16
17
18 # Chargement de l'image à partir d'un site internet
19 response = requests.get("https://tse4.mm.bing.net/th?id=OIP.vUO_4npE9GyhCOZ7BZjUgwHaGL&pid=Api&P=0&w=216&h=181")
20 imgBase = Image.open(BytesIO(response.content))
21 width,height = imgBase.size # Détermine la largeur et hauteur de l'image en pixels
22
23 imgNew = Image.new("RGB",(width,height)) # Création d'une image RGB vierge pour recueillir
24                                           # les modifications apportées
25
26 for i in range(width): # Double boucle permettant de parcourir
27     for j in range(height): # chaque pixel de l'image
28
29         red,green,blue = imgBase.getpixel((i,j)) # Récupère le code RVB du pixel
30         newBlue = blue # Création d'une variable pour stocker une nouvelle
31                        # valeur du canal bleu
32         if newBlue < 240 : # On ne change pas les pixels susceptibles d'être blancs (qualité jpeg)
33             newBlue = 255 - blue # Inversion du canal bleu
34
35         imgNew.putpixel((i,j),(red,green,newBlue)) # Modification du pixel en question
36
37 # Affiche les deux images
38 showImages(imgBase,imgNew)
39
```

Console IPython

Console 1/A



Remarques sur le programme :

- la **création d'une nouvelle image** permet d'éviter à l'image de départ (pomme.jpg) d'être transformée,
- il ne faut pas dépasser la taille de l'image dans les fonctions membres `getpixel()` et `putpixel()` sinon une **exception** sera levée et le programme sera stoppé,
- les variables **red, green et blue** sont des **nombre entiers compris entre 0 et 255 inclus**, utiliser l'instruction `valeur = int(valeur)` si nécessaire pour être sûr que valeur soit un nombre entier.

255 signifie que la couleur primaire est complètement représentée, **0** qu'elle n'existe pas.

Tout nombre non entier ou hors de cet intervalle entraînera aussi la levée d'une exception et l'arrêt du programme.