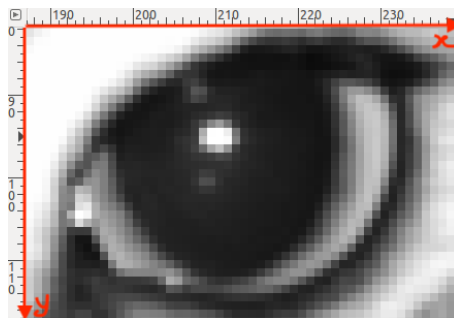


Images numériques

1 Caractéristiques d'une image numérique



Une image en nuances de gris comme celle du chat ci-dessus est composée de pixels (picture element) visibles lorsqu'on a fait un zoom (voir ci-dessous) avec l'aide d'un éditeur d'images comme GIMP



L'intensité de gris pour chaque pixel est un entier entre 0 et 255. 0 correspond au noir et 255 au blanc.

Pourquoi 255 ?

Parce que pour coder une intensité de gris de 0 à 255 on n'utilise qu'un octet c'est à dire huit bits

On retiendra que la taille "moyenne" d'une image en nuances de gris est en octets la largeur par la hauteur d'une image en pixels

C'est une moyenne car les images numériques peuvent avoir différents formats suivant qu'elles sont compressées ou pas

Les formats les plus connus sont le format png (Portable Network Graphics)(compression sans pertes) et jpeg (Joint Photographic Expert Group) (compression avec pertes)

Question 1

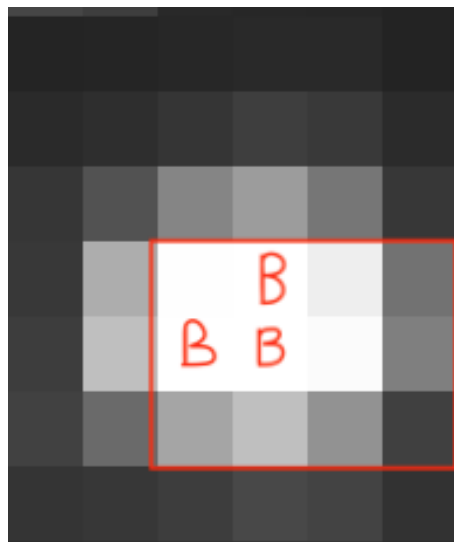
On observe la présence de pixels "blancs" à l'intérieur de l'oeil gauche du chat
En repérant les pixels suivant les axes du repère en rouge donner les coordonnées des pixels "blancs"

```

>>> from PIL import Image
>>> image = Image.open("/Users/vallon/Documents/chat.png")
>>> image.getpixel((209,94))
254
>>> image.getpixel((210,94))
255
>>> image.getpixel((209,95))
255
>>> image.getpixel((210,95))
255
>>> image.getpixel((207,91))
36

```

Nous pouvons observer cela en utilisant la bibliothèque PIL de Python , on fait afficher les intensités des pixels de coordonnées (209,94), (210,94), (209,95) et (210,95) Les 3 pixels "blancs" sont visibles sur le zoom suivant :



Ensuite à l'aide d'un programme Python on fait afficher les intensités des pixels du rectangle rouge

```

from PIL import Image

#on charge l'image du chat
image1 = Image.open("/Users/vallon/Documents/chat.png")

for y in range(94,97):
    print()
    for x in range(209,213):

        print(image1.getpixel((x,y)),end=' ')

```

On obtient :

254 255 238 114
255 255 251 127
165 191 146 64

2 Exemples d'algorithmes de traitement des images

Dans la suite une image numérique, une fois chargée en mémoire **vive** est vue comme un tableau 2D de nombres étiquetée par une variable **img**

La variable **largeur** nous renseigne sur le nombre de **colonnes** du tableau et la variable **hauteur** nous renseigne sur le nombre de **lignes** du tableau

On dispose d'une fonction `lirePixel((x,y))` qui permet d'obtenir l'intensité du pixel situé en (x,y) dans le tableau et d'une fonction `ecrirePixel((x,y),valeur)` qui va remplacer l'intensité en (x,y) en nuances de gris par **valeur**

Quand on écrit `valeur ← img.lirePixel(x,y)` cela signifie que l'on récupère l'intensité en nuances de gris de l'image **img** pour le pixel de coordonnées (x,y) et que ce nombre est ensuite **affectée** à la variable **valeur**

2.1 Négatif d'une image

Dans le premier algorithme qui suit à partir d'une image en nuances de gris on obtient le négatif de l'image

Algorithme 1 : Négatif d'une image en nuances de gris

Données : une image en nuances de gris **img**

Résultat : **img** étiquette le négatif de l'image de départ

```
1 début
2   pour y ∈ [[0; hauteur]] faire
3     pour x ∈ [[0; largeur]] faire
4       valeur ← img.lirePixel(x,y)
5       img.ecrirePixel((x,y), 255 - valeur)
6     fin
7   fin
8   afficher(img)
9 fin
```

1. En Français on parcourt le tableau de la première ligne à la dernière (deux boucles pour imbriquées : **lignes 2 et 3**) et à chaque fois on récupère l'intensité du pixel situé en (x,y) (**ligne 4**) puis on remplace cette valeur par le complémentaire à 255 (**ligne 5**)

2. A la sortie des deux boucles on affiche l'image (**ligne 8**)

Exercice : Exécuter l'algorithme avec l'image **img** = [[200,150,100],[50,60,70]]

2.2 Quantification d'une image

Quantifier est le contraire de **nuancer**. Une image est un ensemble de nuances et il s'agit de diminuer leur nombre. A l'extrême on ne garde que deux nuances, le noir



codé par 0 et le blanc codé par 255.
Proposer un algorithme pour cela



2.3 Détection de contours dans une image

Un contour dans une image est un ensemble de pixels où "a lieu une différence de nuances relativement importante". Par exemple dans l'image du chat , les pixels constituant les moustaches ont des intensités proche de 255 alors que les pixels "voisins" ont des intensités proches de 0, ainsi à l'endroit des moustaches il y a une différence d'intensité relativement importante

Pour détecter les contours on va procéder ainsi :

1. On parcourt l'image comme d'habitude
2. Pour chaque pixel (x,y) on relève l'intensité du pixel en (x,y) int_{NB} , l'intensité du voisin "en dessous" int_{NBB} et "à droite" int_{NBD} **dans le repère de l'image**. Quel sont les coordonnées des voisins en dessous et à droite ?
3. On calcule les valeurs absolues des différences $int_{NB} - int_{NBB}$ et $int_{NB} - int_{NBD}$
4. Si l'une ou l'autre de ces différences est plus grande qu'une valeur fixée au préalable **SEUIL** alors on attribue au pixel (x,y) l'intensité 255 sinon on attribue l'intensité 0

Traduire cette procédure en algorithme

SEUIL = 64



3 Histoire

1. 1826 : Naissance de la photographie argentique
2. 1900 : Photographie en couleurs
3. 1969 : Arrivée des premiers capteurs CDD (Charge Coupled Device)
4. 1975 : Apparition des premiers appareils numériques
5. 2007 : Arrivée du smartphone

4 Enjeux sociétaux

à suivre ...