Python et statistique descriptive: application au traitement d'image

publié le 22/06/2022

Analyser et nettoyer une image grâce aux statistiques

Descriptif :

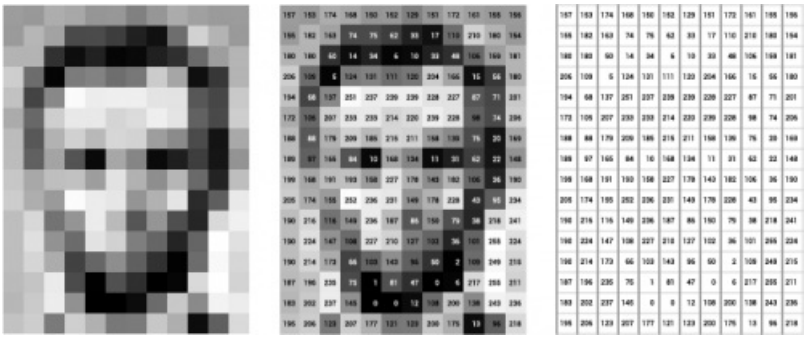
On se propose de montrer que les paramètres statistiques descriptifs (indicateurs de position ou dispersion) permettent de nettoyer une image ou d'extraire des information de l'image. Les analyse et traitements seront exécutés avec des algorithmes écrits en Python dans un notebook (jupyter ou basthon). Les applications jupyter ou basthon sont disponibles dans l'ENT, via l'application Capytale pour Basthon.

Rappelons les capacités et connaissances de statistiques en seconde professionnelle :

Capacités et connaissances en statistiques	
Capacités	**Connaissances**
Comparer et interpréter des séries statistiques à l'aide d'indicateurs de position et de dispersion calculés avec les fonctions statistiques d'une calculatrice ou d'un tableur.	Indicateurs de position : mode, classe modale, moyenne, médiane, quartiles. Indicateurs de dispersion : étendue, écart type, écart interquartile Q 3 – Q 1 .
Construire le diagramme en boîte à moustaches associé à une série statistique avec ou sans TIC. Comparer et interpréter des diagrammes en boîte à moustaches.	Diagrammes en boîte à moustaches.

Qu'est-ce qu'une image numérique ?

Une image en niveaux de gris est un tableau de nombres codés sur 8 bits (il y a $2^8 = 256$ valeurs différentes possibles : de 0 à 255). Chaque nombre code le niveau de gris d'un pixel. La valeur 0 code pour un pixel noir, la valeur 255 code un pixel blanc, les valeurs intermédiaires codent pour des nuances de gris.



Léna

Léna est une image qui a beaucoup servi pour tester des algorithmes d'analyse et de traitement d'image. On se propose de travailler sur cette image avec le regard des statistiques.

lenaBW.png est un fichier accompagnant un notebook disponible dans Capytal. On se propose d'ouvrir le notebook (une connexion à l'ENT est nécessaire) en parallèle de la lecture de cet article.
Comme souvent, l'ajout de quelques modules permet de démultiplier les pouvoirs du langage Python. On commence donc par :

```
import numpy as np
from scipy import ndimage as nd
from imageio import imread
from matplotlib import pyplot as plt
```

et on charge le fichier lenaBW.png par :
`lena = imread('LenaBW.png')`

Profil des niveaux de gris d'une image : densitogramme
Une image en niveau de gris est un tableau de nombres. On peut atteindre chaque pixel un par un, ligne par ligne, ou colonne par colonne ou encore par domaine rectangulaire. Copions une unique ligne de pixels de l'image et traçons la (Une partie du code, voir le notebook, sert également à tracer une version modifiée de lena) :

```
ligne275 = lena[275 , :]
plt.figure(figsize=(12,4))

plt.subplot(121)
plt.title('ligne 275')
plt.imshow(RGB)

plt.subplot(122)
plt.title('Niveaux de gris le long de la ligne 275 (rouge)')
plt.plot(ligne275)

plt.show()
```

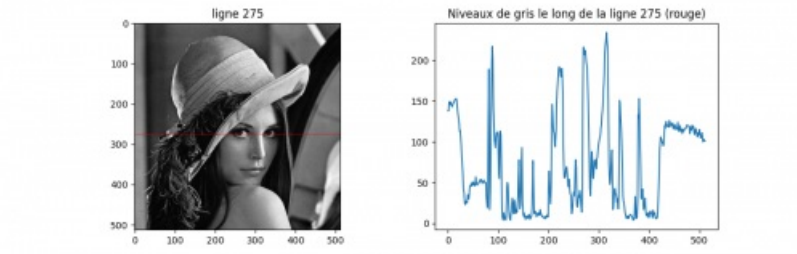


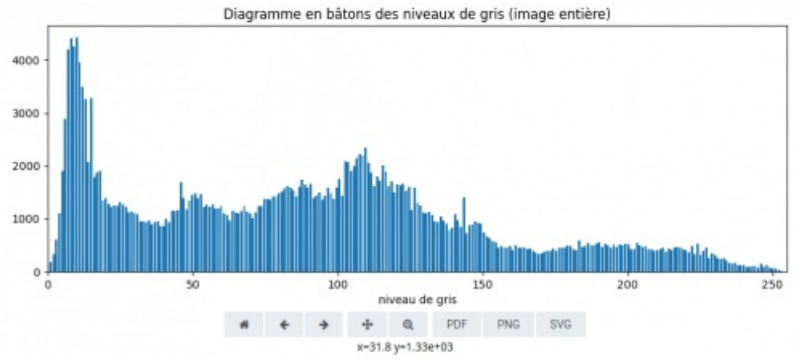
Diagramme en bâtons des niveaux de gris d'une image :
Le diagramme en bâtons (ou l'histogramme) renseigne sur la répartition des niveaux de gris de l'image.
Construisons en un :

- sur l'image entière
- sur une ligne de pixels de l'image (ici la ligne 275, en rouge) les pixels de la ligne 275 de l'image originale de Léna sont dans la variable *densito* (un tableau à une ligne et à 512 colonnes, sauf erreur).

Le code ci-dessous génère et trace le diagramme en bâtons sur l'image entière :

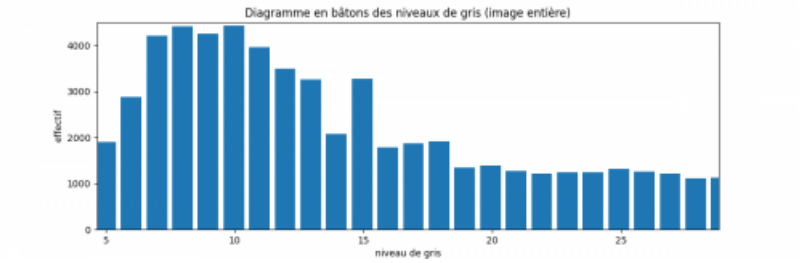
```
#https://datacarpentry.org/image-processing/05-creating-histograms/
effectif, valeur_gris = np.histogram(lena, bins=256, range=(0, 255))

# configure et trace le diagramme en bâtons
plt.figure(figsize=(12,4))
plt.title("Diagramme en bâtons des niveaux de gris (image entière)")
plt.xlabel("niveau de gris")
plt.ylabel("effectif")
plt.xlim([0, 255])  # <- named arguments do not work here

plt.bar(valeur_gris[0:-1], effectif)  # <- or here
plt.show()
```

On obtient le diagramme suivant :



Paramètres de position : le mode

Zoomer interactivement sur la région du "pic le plus haut" du diagramme, donne accès au mode de la série, c'est à dire le niveau de gris modal :



Graphiquement, il y a 4 400 pixels dont le niveaux de gris est égal à 10.

Paramètres de position : la médiane

TODO

Calculons la médiane des niveaux de gris d'une petite portion de 5x5 (lignes 276 à 28, colonnes 274 à 279) de l'œil droit, de manière à pouvoir faire le calcul à la main ou presque :

```
zoom2 = lena[276:281,274:279]
```

Affichons les valeurs des pixels de ce domaine :

```
print("lena")
print(zoom2)
print()
```

Ce qui donne

```
lena
[[201 193 190 175 153]
```

```
[208 198 197 174 136]
[214 198 190 160 117]
[196 180 162 143 118]
[170 163 156 154 144]]
```

Pour calculer à la main cette valeur médiane, transformons le tableau en liste. Ce tableau est de type *numpy.array*, il fournit la méthode *flatten()* pour réaliser cette transformation :

```
array_lena = zoom2.flatten()
print("lena",array_lena)
```

donne :

```
lena [201 193 190 175 153 208 198 197 174 136 214 198 190 160 117 196 180 162
143 118 170 163 156 154 144]
```

Il reste à trier ces valeurs par ordre croissant avec la méthode *sort()* associée à cette classe d'objet (numpy.array)

```
array_lena.sort()
print(array_lena)
```

On obtient :
`Array([117, 118, 136, 143, 144, 153, 154, 156, 160, 162, 163, 170, 174, 175, 180, 190, 190, 193, 196, 197, 198, 198, 201, 208, 214], dtype=uint8)`

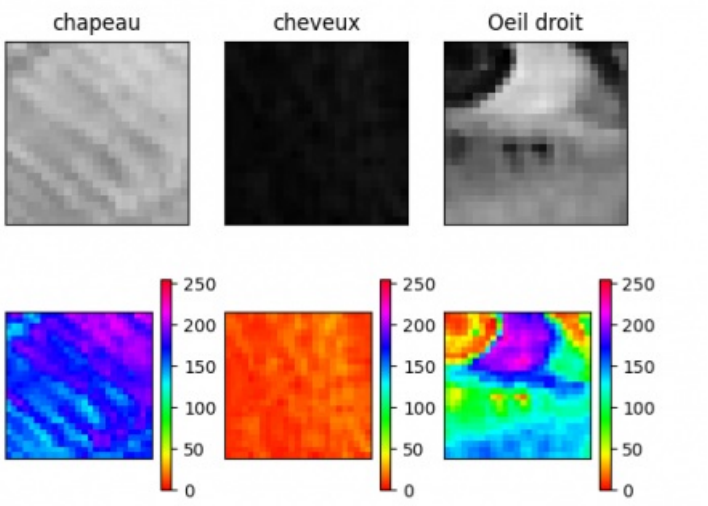
- On détermine le nombre de valeurs dans la liste, c'est à dire la longueur de la liste (le résultat est 25 comme attendu) :
  `len(array_lena)`
- Dans cette situation le rang de la médiane est la 13ième valeur.
- Comment déterminer le rang de cette médiane ? Utiliser une division euclidienne :
  ```
  rang_mediane = 1+len(array_lena)//2
  mediane = array_lena[rang_mediane]
  print(mediane)
  ```

  Le résultat est **160**

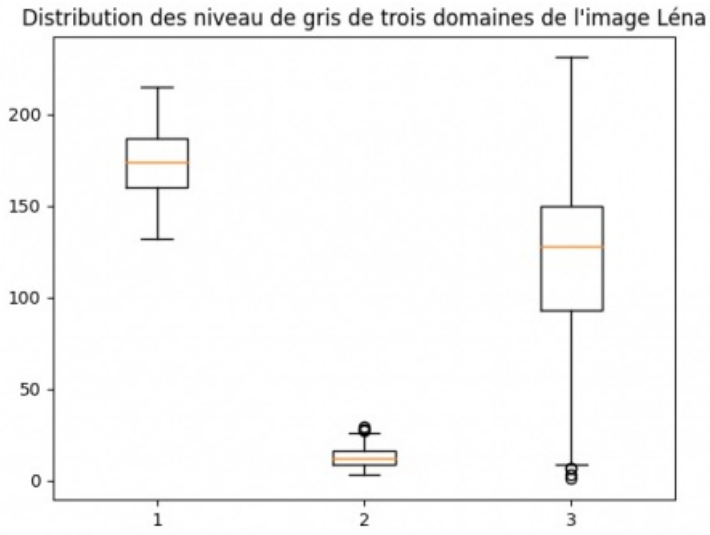
Comparer les statistiques de différents domaines d'une même image :
Construisons les diagrammes en boîte des niveaux de gris de trois domaines de l'image Léna originale.
Réalisons trois copies de dimensions 25x25 de portions de l'image ciblant : le chapeau, les cheveux, la transition claire/sombre de l'œil droit. Ces domaines sont extraits ainsi :

```
lignes_1 = slice(120,145)
colonnes_1 = slice(250,275)
chapeau = lena[lignes_1, colonnes_1 ]

lignes_2 = slice(370,395)
colonnes_2 = slice(180,205)
cheveux = lena[lignes_2, colonnes_2 ]

lignes_3 = slice(270,295)
colonnes_3 = slice(260,285)
oeilDroit = lena[lignes_3, colonnes_3 ]
```

Le premier domaine est composé de pixels des lignes 120 à 145 et des colonnes 250 à 275 de l'image *lenaBW.png*.
Affichons ces domaines, en niveau de gris ou en fausses couleurs :



Puis construisons les diagrammes en boîte correspondant à chaque image :

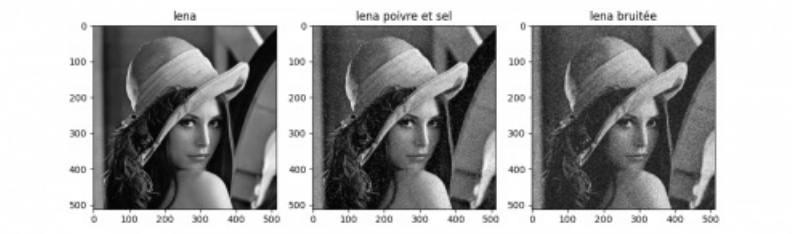


Question : en s'aidant de l'échelle de fausse couleur des domaines *chapeau*, *cheveux*, *oeilDroit*, comment associer chaque image à son diagramme en boîte ?

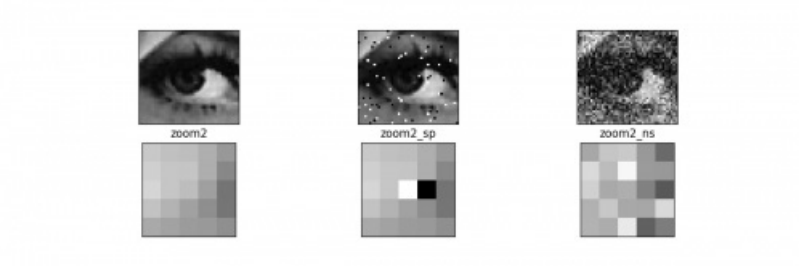
"Débruiter" Léna : le filtrage médian

On se propose de travailler sur trois versions de l'image :

- l'image originale en niveau de gris à gauche
- l'image "poivre et sel" (on a ajouté aléatoirement des pixels noirs ou blancs à l'image)
- l'image bruitée en niveau de gris.
  Ces prétraitement ont été réalisé avec l'application ImageJ.



On extrait la même région dans ces trois images, ce qui revient à zoomer :



Dans basthon, le code python strictement nécessaire qui de permet de charger et d’extraire des domaines des trois images (originale, "poivre et sel", bruitée) est le suivant :

```
saltPepper = imread('LenaSaltPepper.png')
noisy = imread('Lena_noisy.png')

zoom2 = lena[276:281,274:279]
zoom2_sp = saltPepper[276:281,274:279]
zoom2_ns = noisy[276:281,274:279]
```

Affichons les valeurs des pixels de ces domaines :

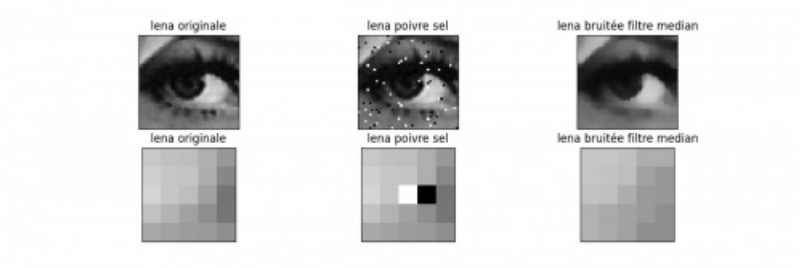
```
print("lena")
print(zoom2)
print()

print("lena poivre sel")
print(zoom2_sp)
print()

print("lena bruitée")
print(zoom2_ns)
print()
```

ce qui donne :

```
lena
[[201 193 190 175 153]
[208 198 197 174 136]
[214 198 190 160 117]
[196 180 162 143 118]
[170 163 156 154 144]]

lena poivre sel
[[201 193 190 175 153]
[208 198 197 174 136]
[214 198 255   0 117]
[196 180 162 143 118]
[170 163 156 154 144]]

lena bruitée
[[168 198 202 153 107]
[218 188 248 155 153]
[209 170 175 130  90]
[178 200 169 167 218]
[179 173 234  97 126]]
```

Filtrage median

le filtrage median d'une image est un algorithme qui balaie une image pixel par pixel. Pour chaque pixel, on calcule la mediane des niveaux de gris sur ce voisinage et on l'attribue au pixel au centre du domaine. Sur un domaine carré de 5x5 le pixel central a 24 voisins, sur un domaine de 3x3, le pixel central a 8 voisins.
Le filtrage median avec un voisinage de 5x5 sur un domaine plus ou moins grand est rendu sur les deux images de droite :



Le filtrage median est réalisé avec une fonction du module ndimage (celui que l'on a importé au début avec le raccourci nd) sur toute l'image :
`lena_median = nd.median_filter(lena, size=(5,5))`

ou sur des parties de celle-ci :

```
zoom_sp_median = nd.median_filter(zoom_sp, size=(5,5))
zoom2_sp_median = nd.median_filter(zoom2_sp, size=(5,5))
```

L'affichage des valeurs de niveaux de gris des domaines de dimension 5x5 (lena, version poivre et sel, et après filtrage) sont donnés :

```
lena
[[201 193 190 175 153]
[208 198 197 174 136]
[214 198 190 160 117]
[196 180 162 143 118]
[170 163 156 154 144]]

lena poivre sel
[[201 193 190 175 153]
[208 198 197 174 136]
[214 198 255   0 117]
[196 180 162 143 118]
[170 163 156 154 144]]

lena filtre median
[[198 198 193 174 174]
[198 196 190 162 153]
[196 193 174 154 153]
[180 170 163 154 144]
[180 170 162 144 144]]
```

Questions pour terminer

- Comment mettre en évidence que la médiane est un paramètre de position peu sensible aux valeurs extrêmes ? (penser à comparer les diagrammes en boîte du même domaine dans chaque image).
- Quel est l'effet du filtre median sur la statistique de l'image ?
    - pourquoi ne pas comparer les profil densitomètriques d'une image et de sa version filtrée ?

    - les effets du filtrage sont-ils visibles dans les diagrammes en boites ?
- Quel est l'effet des dimensions du voisinage utilisé dans le filtre median ?
- A quoi ressemblerait une image si la valeur du pixel central n'était pas le niveau de gris median du voisinage, mais un paramètre de dispersion, comme l'écart type ? En général c'est la variance qui est utilisée (le carré de l'écart-type). Une piste : pour une zone totalement homogène (de même niveau de gris), la variance sera nulle (pas de dispersion des valeurs).