

Niveaux de gris et couleurs

1. Niveaux de gris d'une image monochrome

1.a. Obtention d'une image monochrome

```
Needs['CV'];  
CVOpen[];
```

Lorsqu'on ne souhaite pas exploiter les informations de couleur d'une image, on peut la lire directement sous forme d'une image monochrome :

```
img=CVLoadImage['airplane.jpg',CVLoadImageGrayScale];
```

L'image obtenue comporte 1 couche de 8 bits. Les valeurs des niveaux de gris sont donc comprises entre 0 et 255.

Affichons l'image dans le notebook :

```
array = CVGetImageArray[img,1,1];
```

```
Image[array]
```



1.b. Histogramme des niveaux de gris

L'histogramme des niveaux de gris représente le nombre de pixels ayant une valeur donnée en fonction de cette valeur. Il faut tout d'abord créer l'histogramme :

```
hist=CVCreateHist[{256},CVHistArray,{{0,255}},1];
```

Le premier argument est une liste donnant le nombre de valeurs dans chaque dimension (ici il n'y a qu'une seule dimension). Le troisième argument précise les bornes des valeurs. Pour calculer l'histogramme :

```
CVCalcHist[{img},hist,0,0];
```

On remarque que l'image doit être transmise dans une liste à un seul élément (histogramme 1D).

Les valeurs de l'histogramme peuvent être récupérées sous forme d'une liste :

```
h=CVGetHistArray[hist];
```

Voyons par exemple l'élément 100 de cette liste :

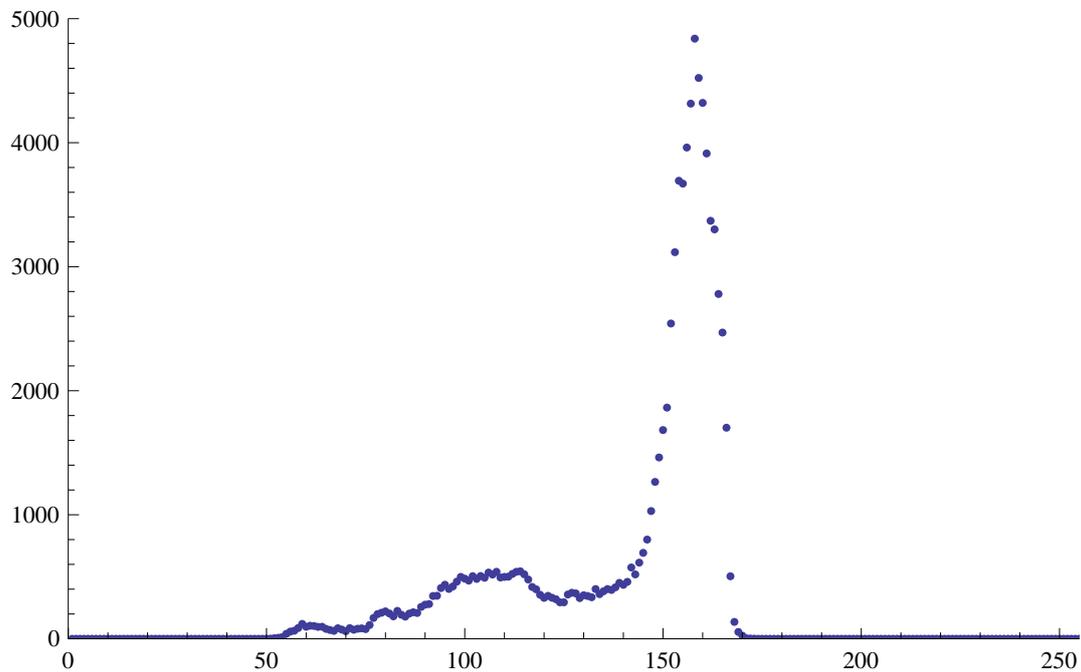
```
h[[100]]
```

```
485.
```

Il y a donc 485 pixels qui ont la 100 ième valeur de l'intervalle 0,255, c'est-à-dire la valeur 101.

La représentation graphique est obtenue par :

```
ListPlot[h,PlotRange->{{0,255},{0,5000}}]
```



La valeur minimale et la valeur maximale de l'histogramme sont obtenues par :

```
minmax=CVGetMinMaxHistValue[hist]
```

```
{{0., 0.}, {4839., 157.}}
```

Le premier doublet donne la valeur minimale (0 pixels) et le niveau correspondant (0, le plus faible). Le second doublet indique la valeur maximale (4839 pixels) et le niveau correspondant (157). Les pixels les plus nombreux sont donc ceux de valeur 157, qui correspondent au maximum visible sur la courbe.

Il est possible de normaliser l'histogramme de telle sorte que la somme de ses valeurs soit égale à une valeur choisie. Par exemple, pour que la somme soit égale à 1.0 :

```
CVNormaliszeHist[hist,1.0];
```

1.c. Ajustement des niveaux de gris

L'ajustement des niveaux de gris permet de modifier la luminosité et/ou le contraste d'une image monochrome. L'image précédente souffre d'un important défaut de contraste. Comme on le constate sur l'histogramme, la valeur la plus élevée des pixels est 170 alors qu'il s'agit du fond neigeux qui devrait être proche de 255. De même, la valeur minimale devrait être proche de 0 et non pas 50.

Pour modifier cette image, on commence par créer une nouvelle image monochrome de la même taille :

```
s=CVGetSize[img];  
img2=CVCreateImage[s,IplDepth8U,1];
```

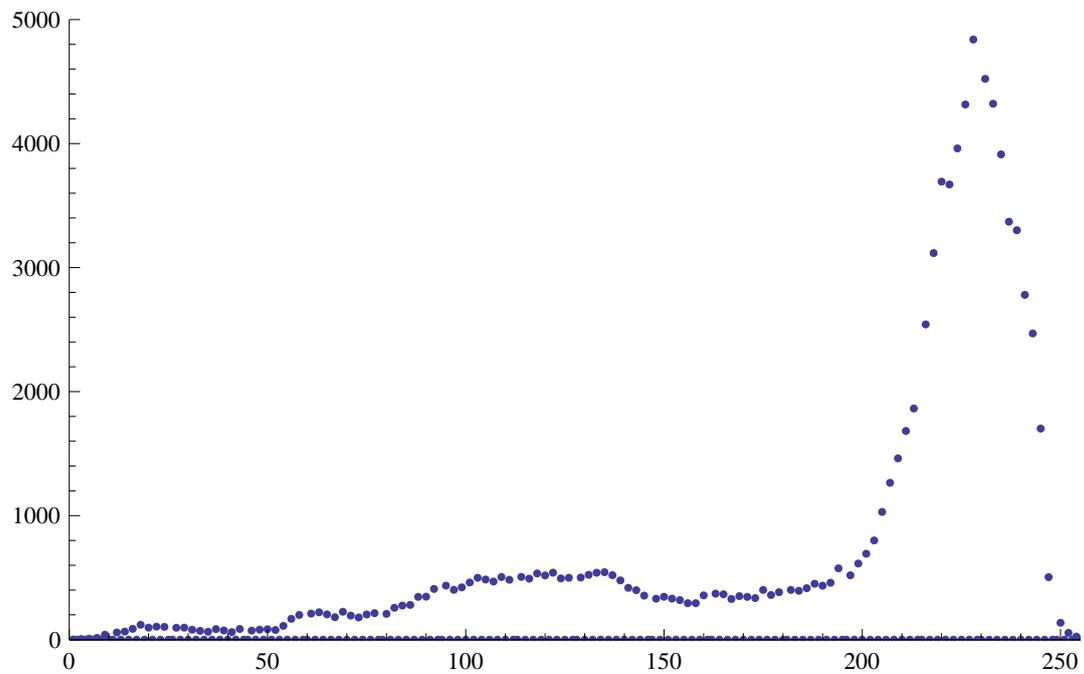
Pour étendre les valeurs de l'image sur toute la gamme possible (0 à 255), il faut multiplier ces valeurs par $255/(170-50)$ puis retranche $50*255/(170-50)$. Ceci est obtenu par la fonction suivante :

```
CVConvertScale[img,img2,255.0/(170-50),-50.0*255.0/(170-50)];
```

Voyons le nouvel histogramme :

```
CVCalcHist[{img2},hist,0,0];  
h=CVGetHistArray[hist];
```

```
ListPlot[h,PlotRange->{{0,255},{0,5000}}]
```



```
Image[CVGetImageArray[img2,1,1]]
```



1.d. Seuillage

Le seuillage permet d'obtenir une image binaire. Si par exemple nous voulons isoler les parties noires de l'avion :

```
img3=CVCreatImage[s, Ip1Depth8U, 1];  
CVThreshold[img2, img3, 50.0, 255.0, CVThreshBinary];
```

```
Image[CVGetImageArray[img3, 1, 1]]
```



2. Images en couleur

2.a. Couches BGR

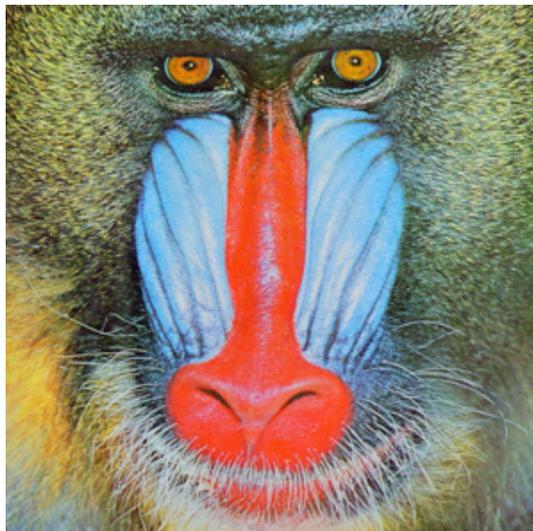
On commence par lire une image en couleur :

```
img4=CVLoadImage['baboon.jpg', CVLoadImageUnchanged];
```

L'image comporte 3 couches (Blue Green Red) de 8 bits chacune. Pour obtenir une image Mathematica, il faut obtenir un tableau de valeurs normalisées et dont l'ordre des couches est inversé (Red Green Blue) :

```
array=CVGetImageArray[img4, 1, -1];
```

```
Image[array]
```



La majorité des opérations de traitement d'image se fait sur des couches individuelles. Pour obtenir les couches B, G et R il faut tout d'abord créer trois images monochromes de même taille que l'image BGR :

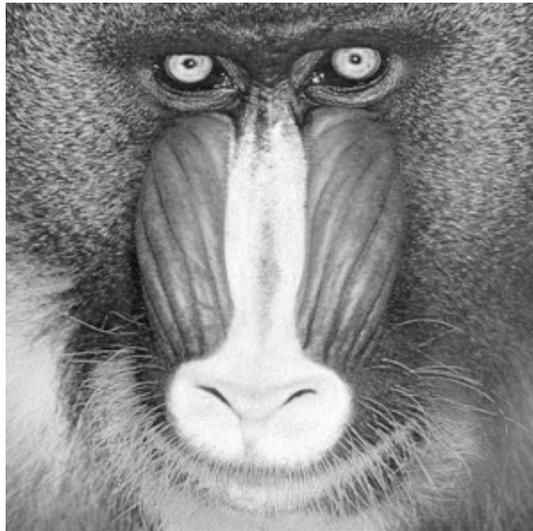
```
s=CVGetSize[img4];  
blue=CVCreateImage[s, IplDepth8U, 1];  
green=CVCreateImage[s, IplDepth8U, 1];  
red=CVCreateImage[s, IplDepth8U, 1];
```

L'image BGR est scindée en trois couches :

```
CVSplit[img4, blue, green, red, 0]
```

Voyons par exemple la couche rouge :

```
Image[CVGetImageArray[red, 1, 1]]
```



2.b. Couches HSV

Les informations de couleurs sont beaucoup plus facilement extraites si l'image est convertie en couches HSV (Hue Saturation Value) :

```
CVCvtColor[img4, img4, CVBGR2HSV]
```

La couche H (Hue) contient l'information de couleur, la couche S (Saturation) contient l'information de saturation des couleurs et la couche V (Value) l'information de luminance.

La séparation en couches s'effectue comme précédemment :

```
hue=CVCreateImage[s, IplDepth8U, 1];  
sat=CVCreateImage[s, IplDepth8U, 1];  
val=CVCreateImage[s, IplDepth8U, 1];  
CVSplit[img4, hue, sat, val, 0]
```

Les valeurs de Hue sont comprises entre 0 et 180 alors que les valeurs de Sat et Val sont comprises entre 0 et 255.

2.c. Histogramme HS des couleurs

Un histogramme 2D Hue-Sat représente le nombre de pixel en fonction de (H,S). On doit tout d'abord créer l'histogramme 2D :

```
histHS=CVCreatHist[{20,20},CVHistArray,{{0,180},{0,255}},1];
```

puis le calculer à partir des couches H et S :

```
CVCalcHist[{hue,sat},histHS,0,0]
```

Voyons les valeurs minimales et maximales :

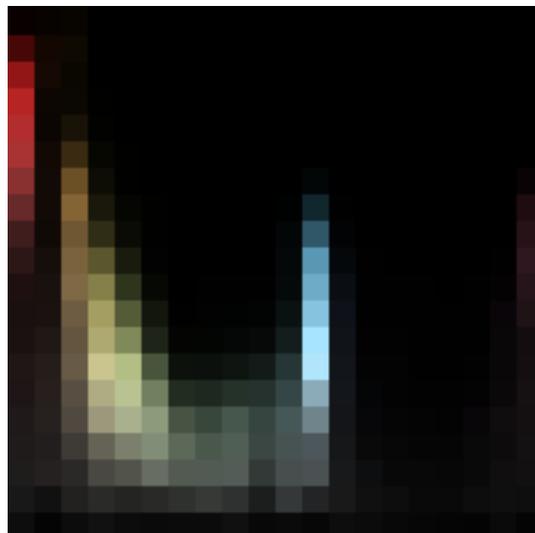
```
minmax=CVGetMinMaxHistValue[histHS]
```

```
{{0., 3., 17.}, {2021., 11., 7.}}
```

Une représentation graphique de l'histogramme peut être obtenue de la manière suivante, sous forme d'une image BGR :

```
histImg=CVGetHueSatHistImage[histHS,10];
```

```
Image[CVGetImageArray[histImg,1,-1]]
```



La valeur de Hue est en abscisse (ici il y a 20 valeurs réparties uniformément entre 0 et 180), la valeur de Sat en ordonnée (origine en bas à gauche). L'intensité du carré est proportionnelle au nombre de pixels. Ici, le maximum de pixels est obtenu pour une couleur bleue ($H=11/20*180, S=7/20*256$)

```
CVClose[]
```