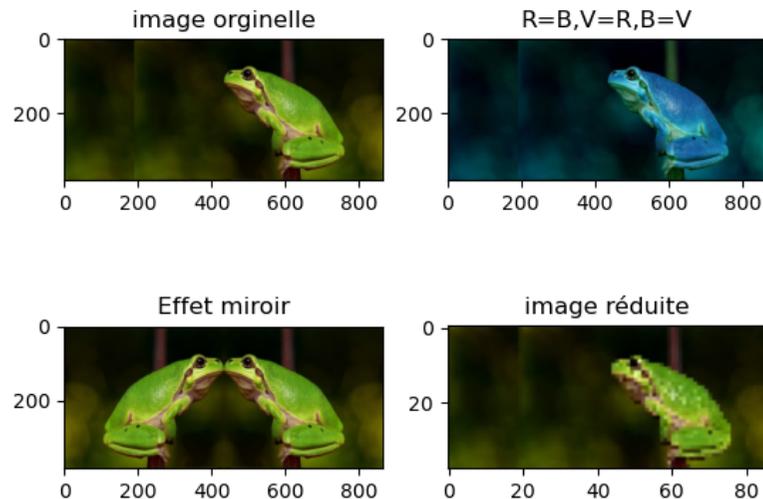


Images en couleur...

Originaires d'Amérique du sud et d'Amérique centrale, les grenouilles vertes mâles Tungara sont connus pour utiliser un cri particulier pour attirer les femelles en vue de l'accouplement. Une étude scientifique a cherché à comparer le pouvoir de séduction des grenouilles mâles qui évoluent à la fois en plein centre-ville de Panama (Panama) et dans ses environs plus ruraux. Selon Wouter Halfwerk, un chercheur participant au projet, " Leur coassement ressemble un peu au bruit que fait un flipper électrique".

Le but de ce TP est d'obtenir les images suivantes :



Les fichiers à utiliser sont `grenouille.jpg` et `TP_image_II.py`.

Q 1 - Calculer le nombre d'octets nécessaires pour coder une telle image. Comparer à la taille du fichier `grenouille.jpg`. Conclure.

1 - Changement de couleur

Le code fourni permet de récupérer sous la forme d'une liste de listes le codage d'une image en RGB (pour Red Green Blue). On rappelle que `img[i][j]` est alors une listes

de 3 valeurs (par ex. `[24,89,156]` codant les niveaux de chaque couleur entre 0 et 255).

Définir une fonction `change_couleur(img)` qui renvoie une image où les codages de chaque couleur on été échangé.

2 - Effet miroir

On cherche à modifier uniquement la moitié gauche de l'image. Proposer une fonction `symetrique` qui retourne une image dont la moitié gauche est le symétrique de la moitié droite.

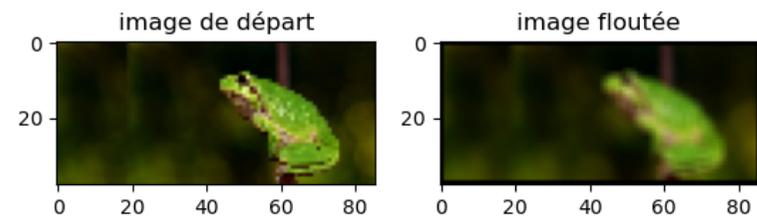
3 - Changement de résolution

Définir une fonction `reduction(img, alpha)` qui retourne une matrice image réduite d'un facteur `alpha` compris entre 0 et 1. Pour contrôler votre fonction, on utilisera un facteur de réduction de `alpha = 0.1` afin d'obtenir l'image pixellisée ci-contre.

Pour aller plus loin...

4 - Floutage

Pour flouter une image, ou au contraire en accentuer les détails, la technique la plus simple consiste à remplacer la valeur de chaque pixel par une valeur calculée sur une petite fenêtre de taille `N` centrée autour de ce pixel.



Q 2 - Définir une fonction `moyenne_couleur(img, i, j, N)` qui renvoie un triplet de valeurs entières correspondant à la moyenne des valeurs de chaque pixel pour chaque couleur.

Pour contrôler votre fonction, on travaillera sur l'image réduite notée `img_4`. Vous devriez obtenir un résultat similaire :

```
1 >>>moyenne_couleur(img_4,10,12,5)
2 [20, 20, 1]
```

Q 3 - Proposer une fonction `flou(img,N)` qui renvoie une image floutée par moyennage uniforme sur une fenêtre de taille `N`.

5 - Rotation

L'image `terre.jpg` correspond à la terre vue du pôle. On cherche à représenter la terre à différents instants par une rotation d'un angle `theta`.

Q 4 - Définir le centre de l'image de coordonnées `i_c,j_c`

Q 5 - Définir la transformation permettant d'effectuer une rotation de la terre d'un angle `theta`. On montrera que la loi s'écrit :

```
1 i2 = i_c + int(cos(theta)*(i-i_c)+sin(theta)*(j-j_c))
2 j2 = j_c+ int(cos(theta)*(j-j_c)-sin(theta)*(i-i_c))
```

Q 6 - Proposer un code permettant de représenter la terre pour différents angles de rotation.

