

✓ Ds ## 2. Partie 1 : Bruitage d'une Image  
# 2.1 Chargement des bibliothèques

```
import numpy as np
import cv2 as cv
import random
from matplotlib import pyplot as plt
from skimage.util import random_noise
```

✓ Ds [4] # 2.2 Lecture et affichage d'une image

```
img = cv.imread('cat.png')
img_rgb = cv.cvtColor(img, cv.COLOR_BGR2RGB)

plt.imshow(img_rgb)
plt.title("Image originale")
plt.axis("off")
plt.show()
```



Image originale



✓ Ds # 2.3 Conversion en niveaux de gris

```
image_gris = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

plt.imshow(image_gris, cmap='gray')
plt.title("Image en niveaux de gris")
plt.axis("off")
plt.show()
```



Image en niveaux de gris



```

✓ 0s  # 2.4 Ajout de bruit "sel et poivre"
def add_noise(img, proportion):
    row, col = img.shape
    noisy_img = img.copy()
    num_pixels = int(proportion * row * col)

    for _ in range(num_pixels // 2):
        x, y = random.randint(0, row - 1), random.randint(0, col - 1)
        noisy_img[x, y] = 255 # Sel

    for _ in range(num_pixels // 2):
        x, y = random.randint(0, row - 1), random.randint(0, col - 1)
        noisy_img[x, y] = 0 # Poivre

    return noisy_img

image_noisy = add_noise(image_gris, 0.02)

plt.imshow(image_noisy, cmap='gray')
plt.title("Image avec bruit sel et poivre")
plt.axis("off")
plt.show()

```



Image avec bruit sel et poivre



```

✓ [7] ## 3. Partie 2 : Filtrage de l'Image
# 3.1 Filtre Médian

img_median3 = cv.medianBlur(image_noisy, 3)
img_median5 = cv.medianBlur(image_noisy, 5)
img_median7 = cv.medianBlur(image_noisy, 7)

```

### Interprétation :

- Plus la taille du noyau augmente, plus l'image est lissée.
- Le filtre médian est très efficace pour supprimer le bruit sans trop flouter les contours.

```
▶ # 3.2 Filtre Gaussien
img_gaussian = cv.GaussianBlur(image_noisy, (5, 5), 0)
```

### Interprétation :

- Atténue le bruit mais peut laisser passer certains pixels corrompus.
- Meilleur pour le bruit gaussien que le sel et poivre.

```
[9] # 3.3 Filtrage Linéaire (moyenne simple)
img_linear = cv.blur(image_noisy, (5, 5))
```

### Interprétation :

- Réduction générale du bruit.
- Lisse également les détails fins de l'image.

```
## 4. Affichage Comparatif

fig, axs = plt.subplots(1, 4, figsize=(15, 5))
axs[0].imshow(image_noisy, cmap='gray')
axs[0].set_title("Bruitée")
axs[1].imshow(img_median5, cmap='gray')
axs[1].set_title("Médian 5x5")
axs[2].imshow(img_gaussian, cmap='gray')
axs[2].set_title("Gaussien 5x5")
axs[3].imshow(img_linear, cmap='gray')
axs[3].set_title("Linéaire 5x5")

for ax in axs:
    ax.axis("off")
plt.show()
```

### Interprétation :

- Le filtre médian est clairement le plus adapté au bruit sel et poivre.
- Le filtre gaussien atténue globalement le bruit mais reste moins efficace.
- Le filtre linéaire est simple mais moins précis.

# Exercice Complémentaire – Impact de l’Intensité du Bruit "Sel et Poivre"

```

✓ [11] # 1. Ajouter du bruit avec différentes intensités
0s
noise_levels = [0.01, 0.05, 0.10]
noisy_images = [add_noise(image_gris, p) for p in noise_levels]

✓ [12] # 2. Appliquer les trois filtres à chaque image bruitée
0s
results = []

for noisy in noisy_images:
    median = cv.medianBlur(noisy, 5)
    gaussian = cv.GaussianBlur(noisy, (5, 5), 0)
    linear = cv.blur(noisy, (5, 5))
    results.append((noisy, median, gaussian, linear))

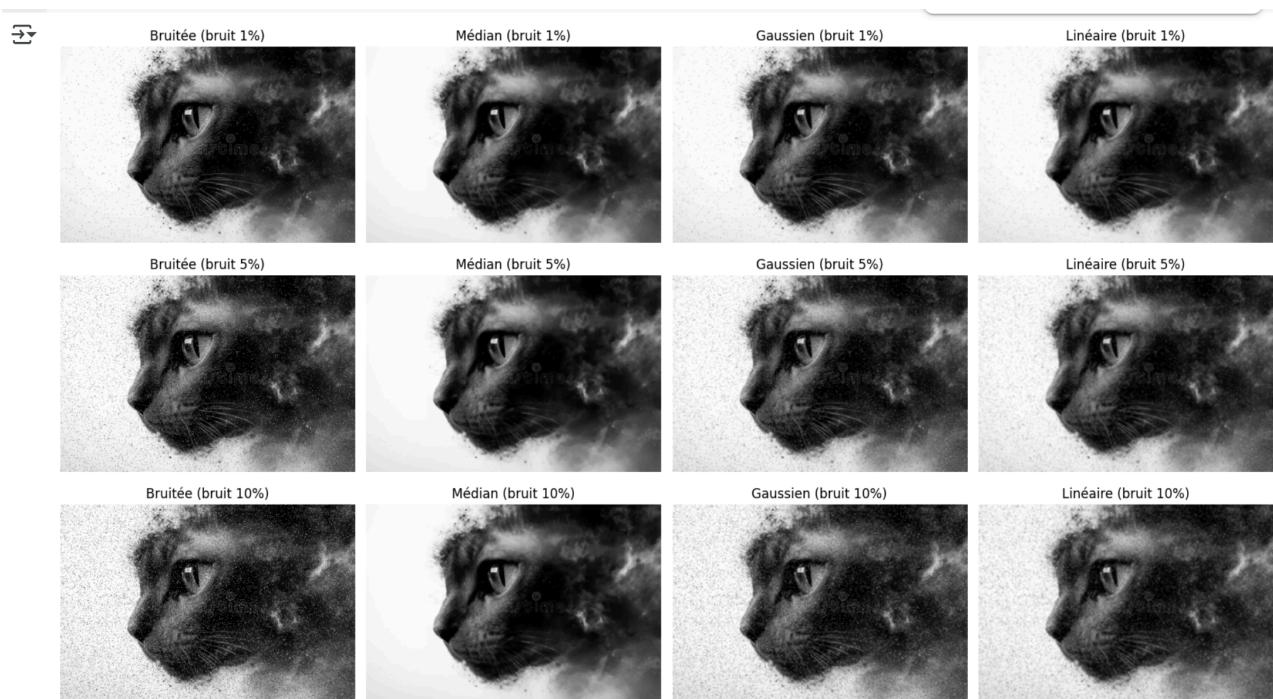
✓ [2s] # 3. Affichage des résultats
# 3. Affichage des résultats
2s
titles = ['Bruisée', 'Médian', 'Gaussien', 'Linéaire']
levels = ['1%', '5%', '10%']

for i, (noisy, median, gaussian, linear) in enumerate(results):
    fig, axs = plt.subplots(1, 4, figsize=(16, 4))
    imgs = [noisy, median, gaussian, linear]

    for j in range(4):
        axs[j].imshow(imgs[j], cmap='gray')
        axs[j].set_title(f'{titles[j]} (bruit {levels[i]})')
        axs[j].axis('off')

    plt.tight_layout()
    plt.show()

```



#### 4. Interprétation des résultats

- Niveau de bruit 1% :
  - Tous les filtres réussissent à corriger la majorité du bruit.
  - Le filtre médian conserve le plus de détails.
- Niveau de bruit 5% :
  - Le filtre médian reste performant.
  - Le filtre gaussien commence à lisser les bords.
  - Le filtre linéaire devient moins efficace : floutage excessif.
- Niveau de bruit 10% :
  - Le filtre médian résiste encore mais les détails sont altérés.
  - Le filtre gaussien et linéaire laissent apparaître plus de bruit et floutent beaucoup.