

# La Compression JPEG

Analyse de la DCT et de la Quantification

BEN Romain  
LECUREUR Evrard  
SAITOUT Zouhair

# Objectifs et Chaîne de Traitement

**Problématique :** Comment réduire la taille d'une image sans perdre l'information essentielle perçue par l'œil ?

- **Méthodologie :**

- Transformation fréquentielle par blocs  $8 \times 8$  via la DCT-II
- Application d'une matrice de **quantification**  $Q$  modulée par un facteur  $\alpha$
- Analyse de la conservation des coefficients (basses vs hautes fréquences)

- **Résultat attendu :**

- Une implémentation réelle stockée en format CSR (.npz) pour prouver le gain de mémoire

## Qualité Image ?



Image originale



Image compressée  
( $\alpha = 1$ )



Image originale

Découpage

Transformation DCT

Quantification  $Q$

Stockage



DCT

Passage du domaine spatial au **domaine fréquentiel**



Quantification  $Q$

Réduction selon la perception humaine  
(etape clé de la compression)



Stockage Sparse CSR

Optimisation de la mémoire, stockage des termes non nuls

- Image divisée en blocs  $8 \times 8$  *pixels*

## Transformée cosinus discrète (DCT) : conversion de l'espace en fréquence

La DCT applique un changement de base orthogonale sur des blocs 8×8, faisant passer l'image du domaine spatial au domaine fréquentiel.

$$D = PMP^T$$

où P est la matrice orthogonale des coefficients de la DCT

$$P_{i,j} = \frac{1}{2}C_j \cos\left(\frac{(2i+1)j\pi}{16}\right), \quad C_0 = \frac{1}{\sqrt{2}}, \quad C_j = 1 \quad (j \geq 1)$$

Cette écriture est équivalente à :

$$D_{k,l} = \frac{1}{4}C_k C_l \sum_{i=0}^7 \sum_{j=0}^7 M_{i,j} \cos\left(\frac{(2i+1)k\pi}{16}\right) \cos\left(\frac{(2j+1)l\pi}{16}\right)$$

# Quantification et impact du facteur $\alpha$

Image originale



JPEG  $\alpha = 1$   
Taux de conservation = 10.75 %



JPEG  $\alpha = 5$   
Taux de conservation = 3.91 %



JPEG  $\alpha = 20$   
Taux de conservation = 1.46 %



$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 13 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

# Filtrage Hautes et Basses fréquences

Conservation (9.70%)



**Action :** On met que des petits coef dans le bas-droite de la matrice Q et des gros dans le reste.

**Résultat :** Perte de la luminosité (DC) et des couleurs.

**Constat :** Seuls les contours subsistent

**Verdict :** Information insuffisante : **l'image est inexploitable.**

Conservation (9.98%)



**Quantification :** On met que des petits coefs dans le coin haut-gauche de la matrice Q et des gros sur le reste

**Constat :** L'essentiel de l'information (couleurs, formes) est préservé.

**Verdict :** C'est la base de la compression : **l'image reste lisible.**



# Filtrage du bruit en compression JPEG

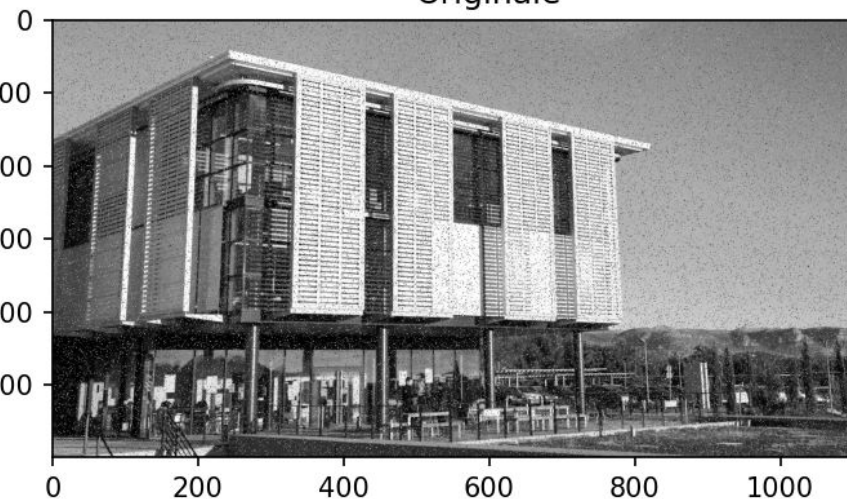
Le **JPEG agit comme un filtre passe-bas**, atténuant le bruit des hautes fréquences. Cependant, ce filtrage entraîne un **flou des contours** et une perte de netteté. Le compromis entre réduction du bruit et préservation des détails dépend du facteur de quantification choisi.

"Test sur bruit 'Poivre et Sel' : Correspond à des variations très rapides (hautes fréquences)

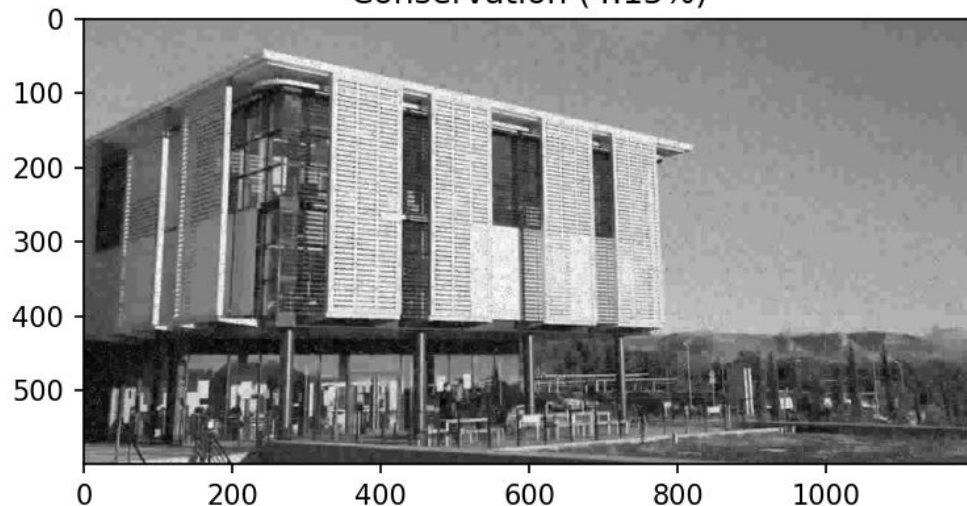
Nettoyage : Suppression du canal Alpha (transparence) pour optimiser les calculs.

Verdict : Le JPEG n'est pas un outil de débruitage robuste ; une forte compression rend les contours flous

Originale



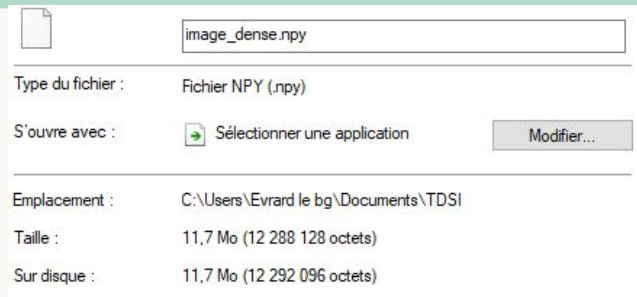
Conservation (4.15%)



# Optimisation via le format de stockage CSR

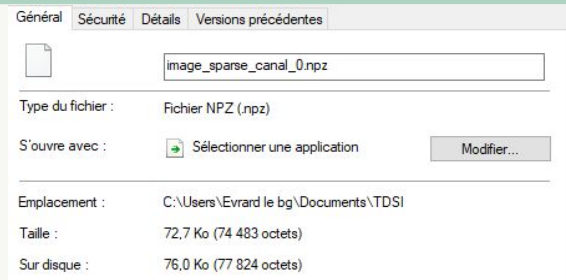
**12 Mo**

Ici on stocke l'objet python en matrice qui représente l'image, c'est à dire les 3 matrices en format dense , on peut retrouver cet espace disque (  $640 * 800 * 8 * 3 / 1024 * 2 = 11,7$  )



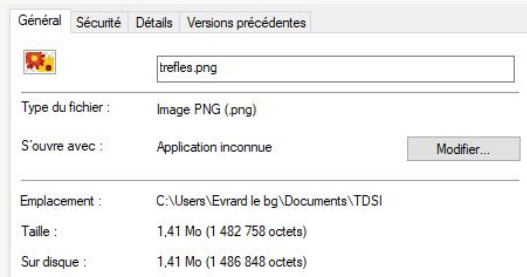
**217 Ko ( taille des 3 matrices combinées )**

Utilisation du format CSR (*Compressed Sparse Row*) de l'image compressée pour ne stocker que les coefficients non nuls.  
( enregistrement format npz )



**1,4 Mo**

Le poids de l'image est donc environ 6 fois supérieur avant la compression sachant que pour cette compression on a utilisé Q normal pour la compression.



# Points clés à retenir de l'analyse de compression JPEG

- **L'œil humain au cœur du JPEG :**
  - Le standard exploite les limites de la perception.
  - La **DCT** isole les fréquences et la **quantification** sacrifie les détails invisibles pour réduire l'information.
- **Performance du stockage CSR :**
  - Démonstration de l'efficacité du format creux : stockage uniquement des coefficients non nuls.
  - **Résultat majeur** : Poids divisé par **6** par rapport au PNG et par **55** par rapport au format dense.
- **Le compromis Qualité/Poids :**
  - Le paramètre  $\alpha$  est le curseur fondamental de la compression.
  - **Limite identifiée** : Le JPEG filtre le bruit "poivre et sel" (effet passe-bas) mais ne remplace pas un vrai débruitage (apparition d'artéfacts de bloc).