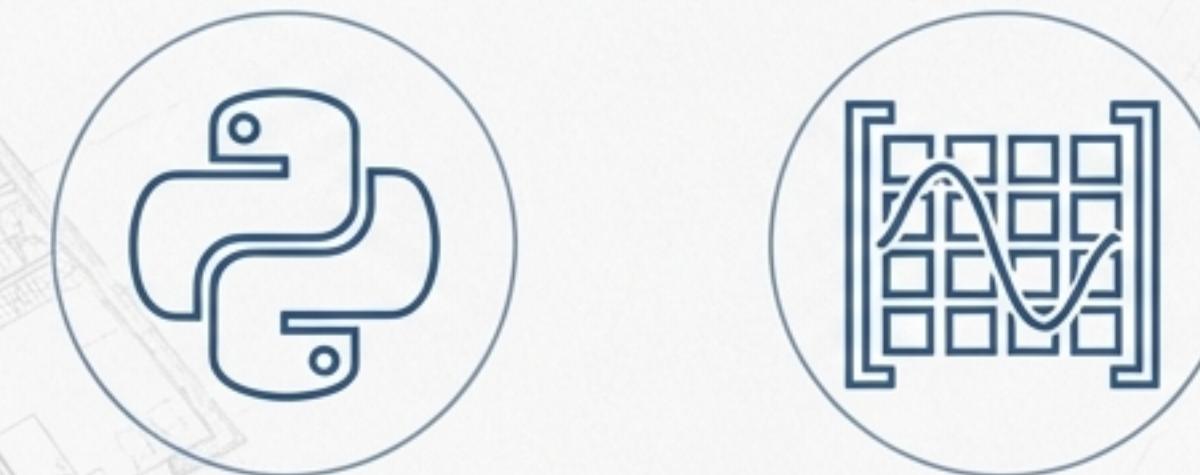


La Boîte à Outils de l'Architecte d'Images

Construction et Analyse d'une Bibliothèque de
Traitement d'Images Numériques

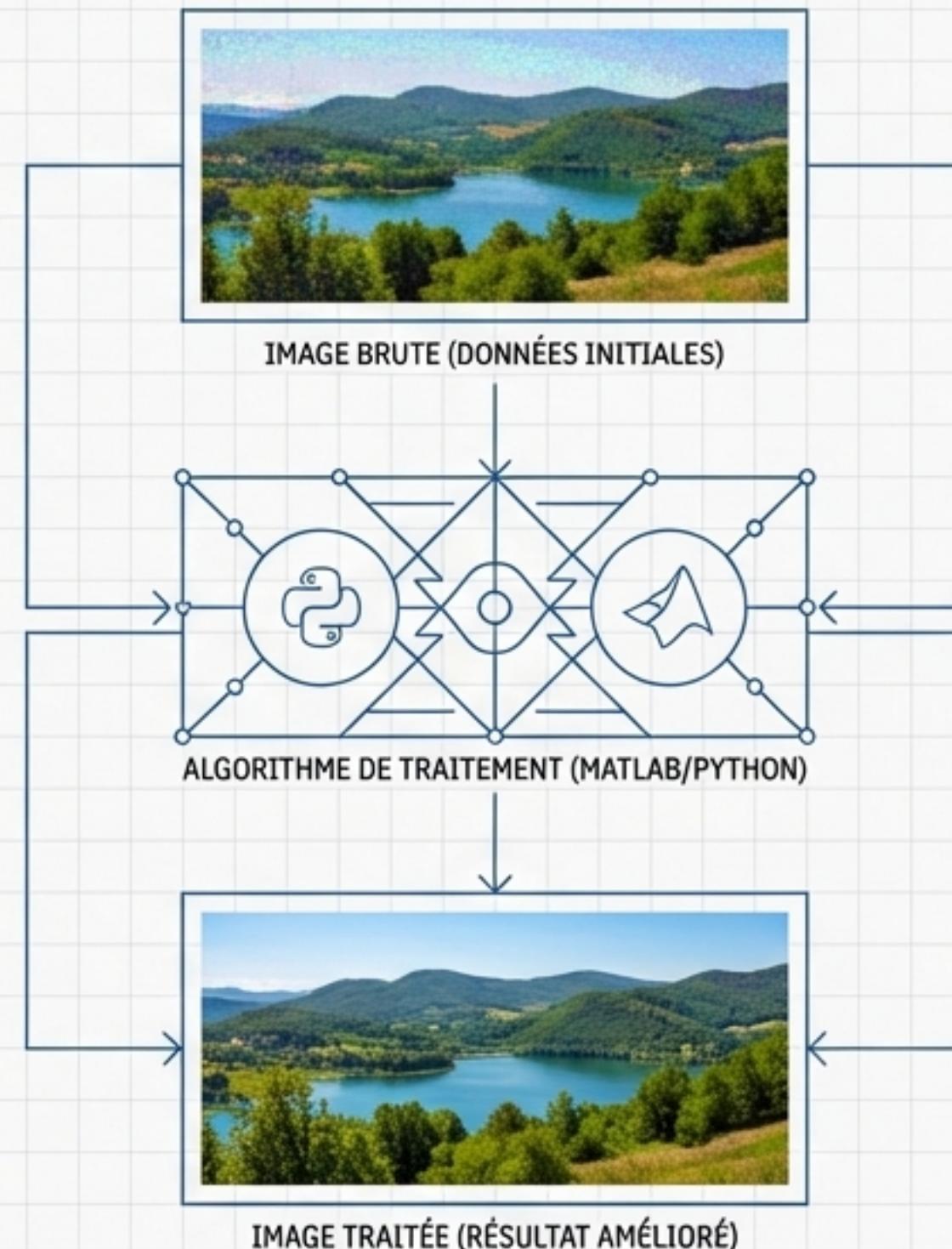


Projet réalisé par : Elghazi Haitam, Elmoutaouakkil Noureddine, Enfissi Monssif, Abderrahmane El-Fennane, Charaf Es-Sail.
Encadré par : Pr. Hafiane & Pr. Aboutabit.
Année 2025-2026.

Le Défi : Maîtriser la Représentation Visuelle

Ce projet vise à maîtriser la représentation des images et leurs opérations élémentaires via une étude comparative entre deux environnements de référence : MATLAB et Python.

L'objectif est de consolider les connaissances théoriques par une implémentation pratique, créant une bibliothèque de fonctions fondamentales pour le traitement d'images.



Les Fondations : Le Codage des Images Numériques

Une image numérique est une matrice de pixels. La valeur de chaque pixel définit sa nature et sa couleur.



Noir et Blanc

0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0

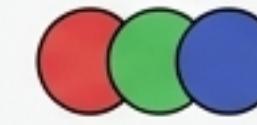
Valeurs binaires (0 pour noir, 1 pour blanc). La forme la plus simple de représentation.



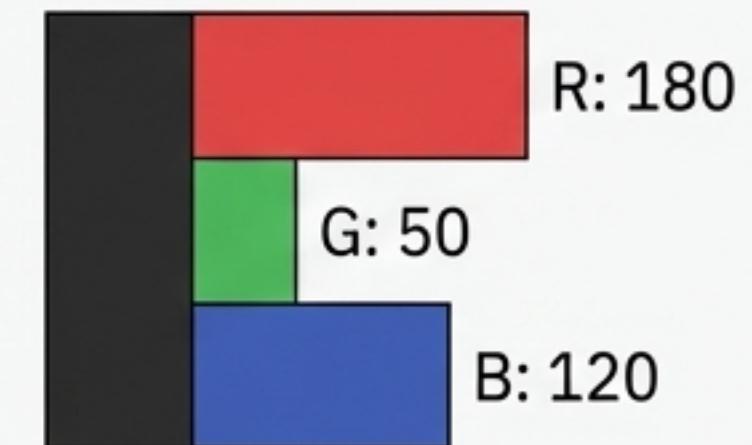
Niveaux de Gris

128	45	210	10
100	150	20	180
230	60	90	130
50	200	80	170

Valeurs d'intensité entre 0 (noir) et 255 (blanc). Capture la luminance sans information de couleur.



RGB (RVB)



Trois composantes (Rouge, Vert, Bleu) par pixel, chacune de 0 à 255. Permet de représenter une large gamme de couleurs.

Outil N°1 : Les Opérations Fondamentales (Entrée/Sortie)

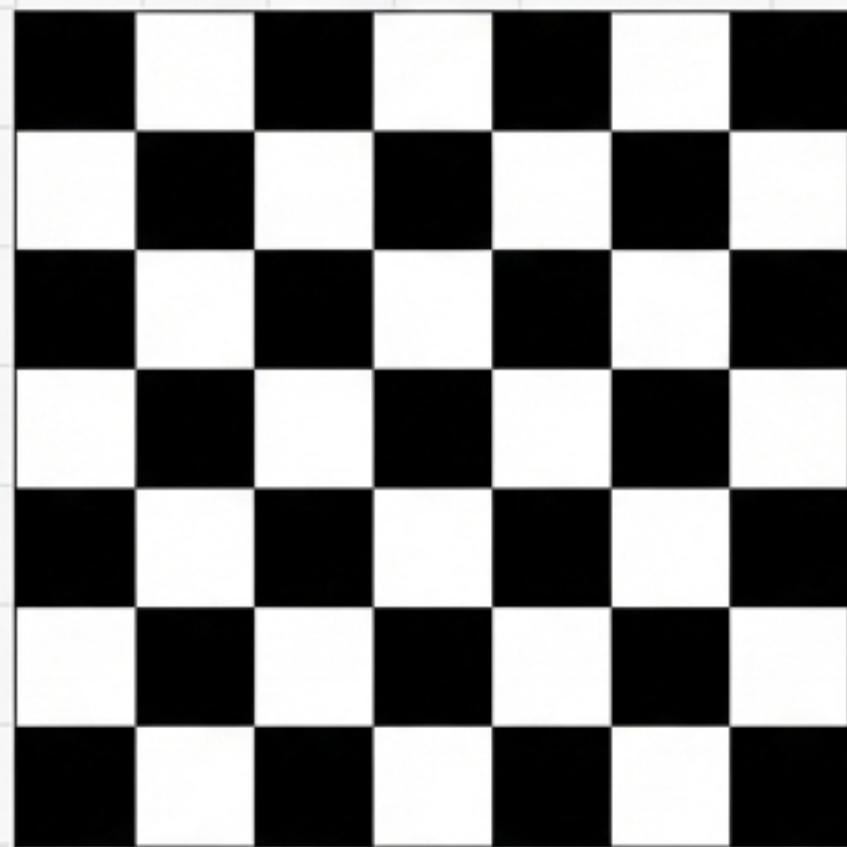
Toute manipulation commence par la capacité à charger, visualiser et sauvegarder une image. Ces fonctions constituent la base de notre bibliothèque.



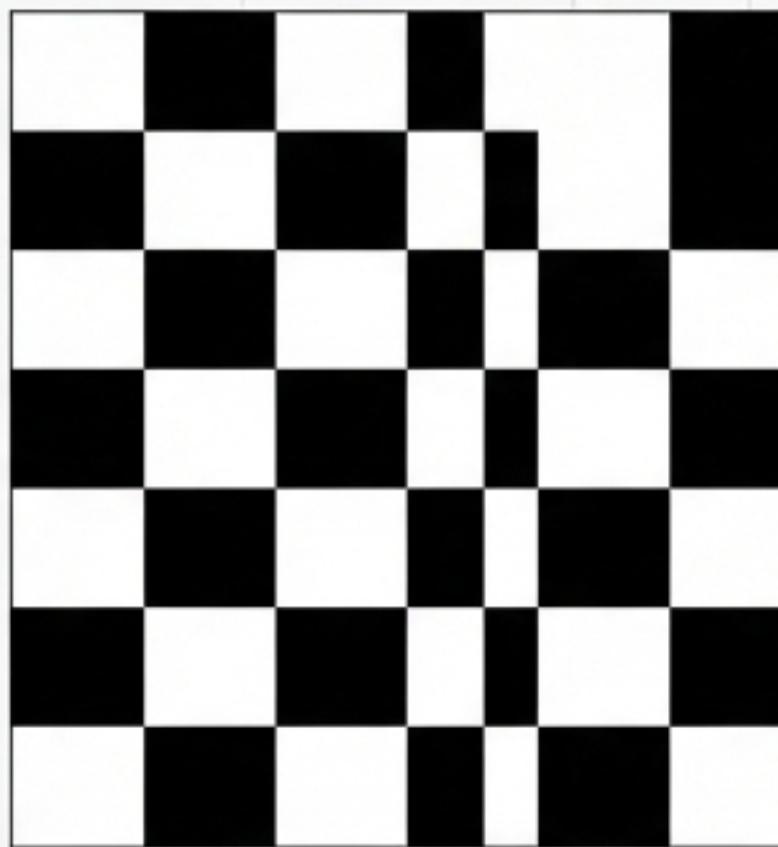
- `ouvrirImage(chemin)` : Lit une image depuis le disque et la charge en mémoire sous forme de matrice.
- `AfficherImg(img)` : Affiche une image à l'écran, sans axes pour une meilleure lisibilité.
- `saveImage(img)` : Enregistre une image traitée sur le disque dur, conservant ainsi les résultats.

Outil N°2 : La Crédation en Noir et Blanc

Ces fonctions permettent de générer des images binaires à partir de zéro et d'effectuer des transformations de transformations de base.



Résultat de 'creerImgBlancNoir'



Résultat de 'negatif'

`image_noire(h, 1)` & `image_blanche(h, 1)` : Génèrent des images entièrement noires (pixels à 0) ou blanches (pixels à 1).

`creerImgBlancNoir(h, 1)` : Crée un motif de damier en alternant les pixels noirs et blancs selon la formule `(i+j) mod 2`.

`negatif(img)` : Inverse les valeurs des pixels (0 devient 1, 1 devient 0) pour créer le négatif de l'image.

Outil N°3 : L'Analyse en Niveaux de Gris

Au-delà de la manipulation, il est crucial d'analyser les caractéristiques radiométriques d'une image. Ces fonctions calculent des statistiques essentielles.



`luminance(img)` :

Calcule la clarté globale de l'image (moyenne de tous les pixels).

`contraste(img)` : Mesure la dispersion des niveaux de gris (variance des pixels). Un contraste élevé indique de fortes variations d'intensité.

`profondeur(img)` :

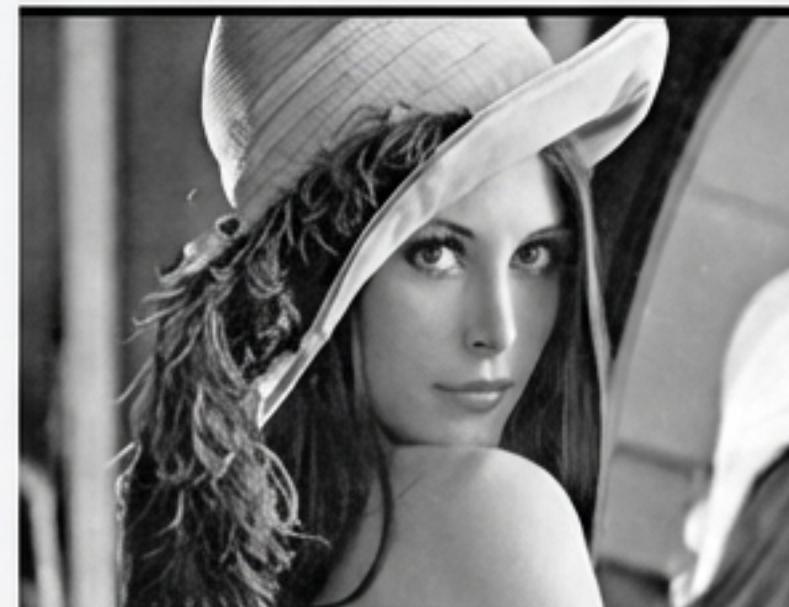
Retourne la valeur maximale d'un pixel, indiquant l'intensité la plus élevée présente.

Outil N°4 : Les Transformations Géométriques

Cet ensemble d'outils permet de manipuler la structure spatiale des images en niveaux de gris.

Inversion Tonale

`'inverser(img)'` (255 - pixel)



Symétrie Axiale

`'flipH(img)'` & `'flipV(img)'`



Concaténation
Horizontale

`'poserH(img1, img2)'`



Concaténation
Verticale

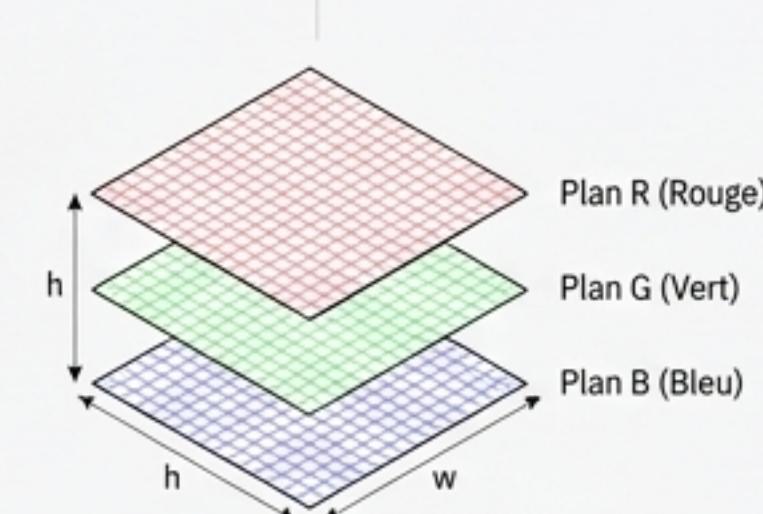
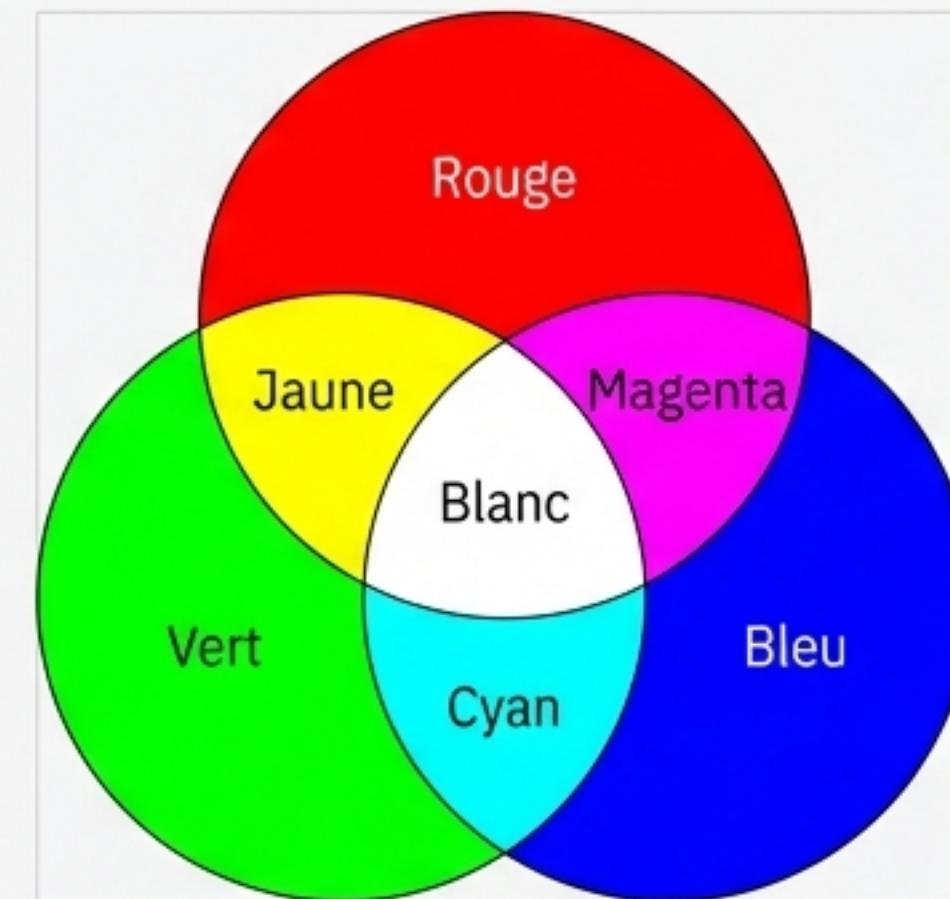
`'poserV(img1, img2)'`



Outil N°5 : La Manipulation des Images RGB

Les images RGB permettent de représenter une large gamme de couleurs proches de la perception humaine. Elles sont stockées sous forme de matrices tridimensionnelles (hauteur \times largeur \times 3).

- Chaque couleur est une combinaison de trois composantes : Rouge, Vert, Bleu.
- Chaque composante est codée sur 8 bits (0-255), permettant plus de 16 millions de couleurs.
- Un pixel est un triplet (R, G, B).

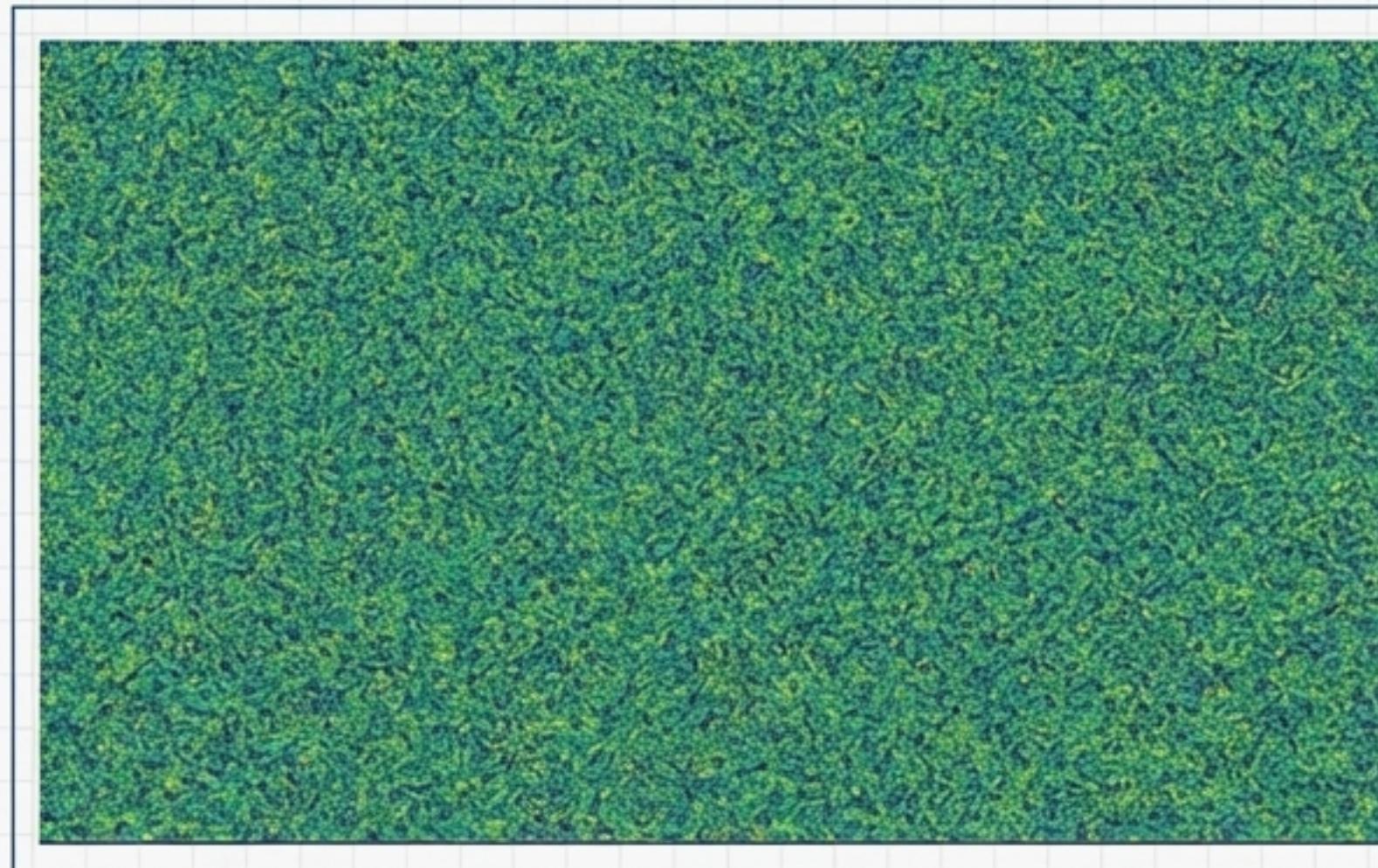


Outils RGB : Initialisation et Transformation

Les fonctions suivantes permettent de générer des images couleur et d'appliquer des symétries sans altérer les informations chromatiques.

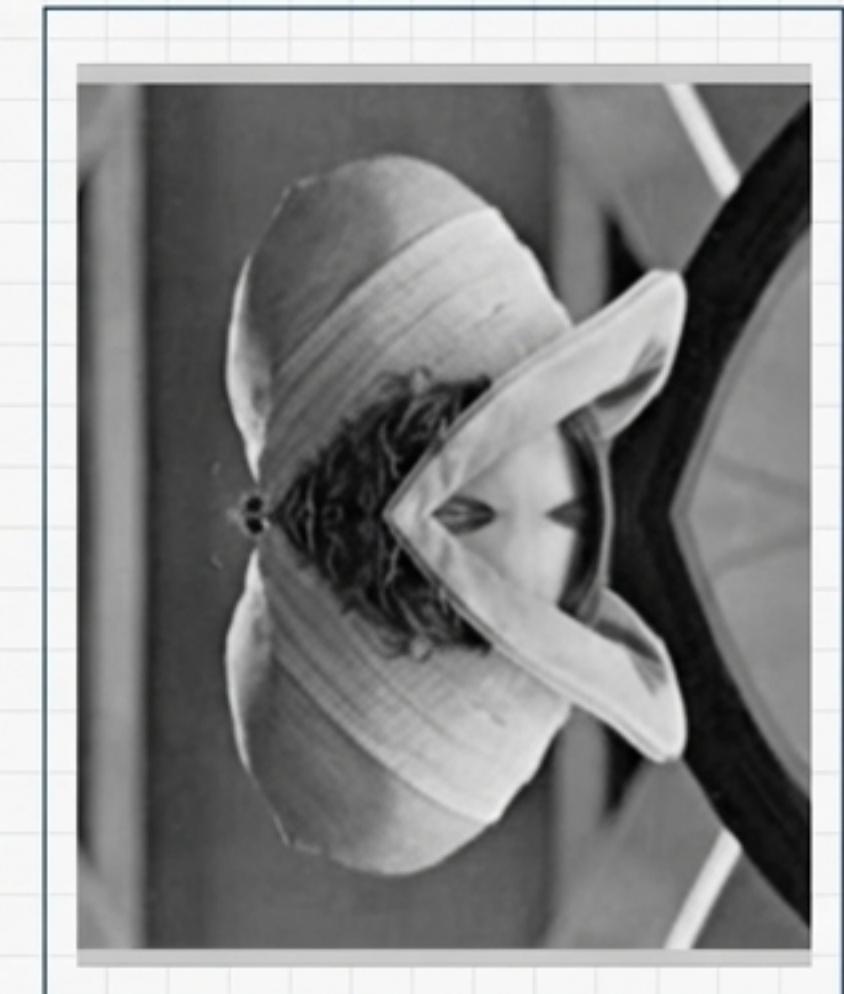
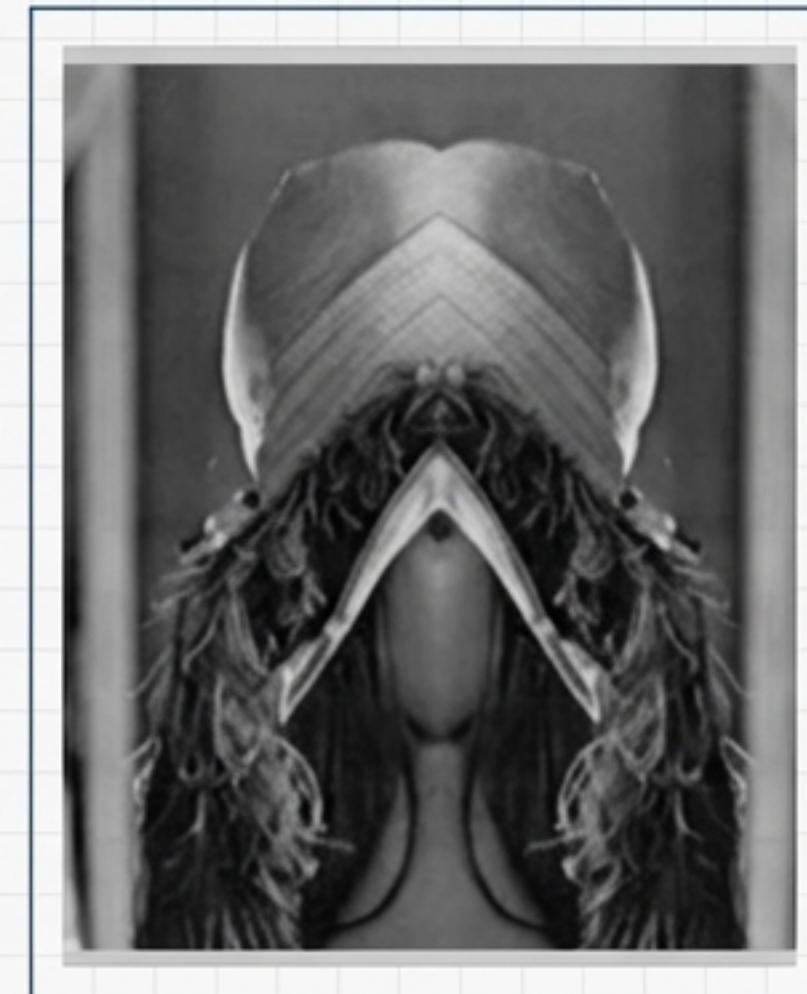
1. Crédit

`'initImageRGB(h,l)'` : Initialise une image avec des pixels de couleur aléatoire, où chaque composante (R, G, B) est une valeur entre 0 et 255.



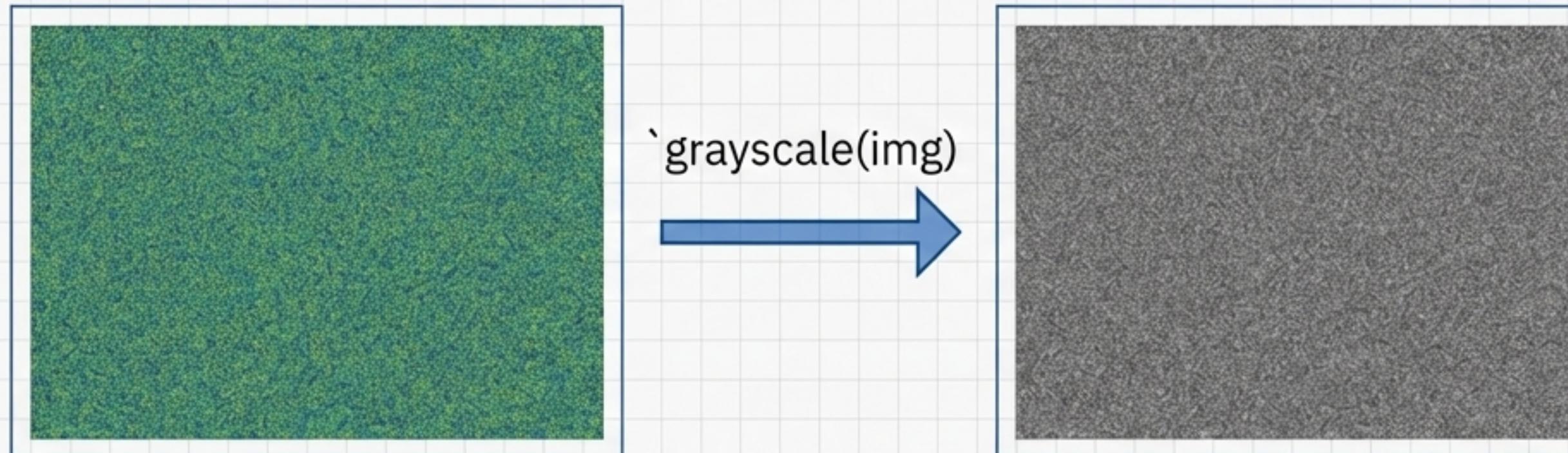
2. Transformation

`'symetrieH(img)'` & `'symetrieV(img)'` : Appliquent une symétrie par rapport à l'axe vertical ou horizontal, comme pour les niveaux de gris.



Outil de Conversion : De la Couleur aux Niveaux de Gris

Une opération essentielle est la conversion d'une image RGB en niveaux de gris, réduisant l'information à la seule luminance tout en conservant les détails.

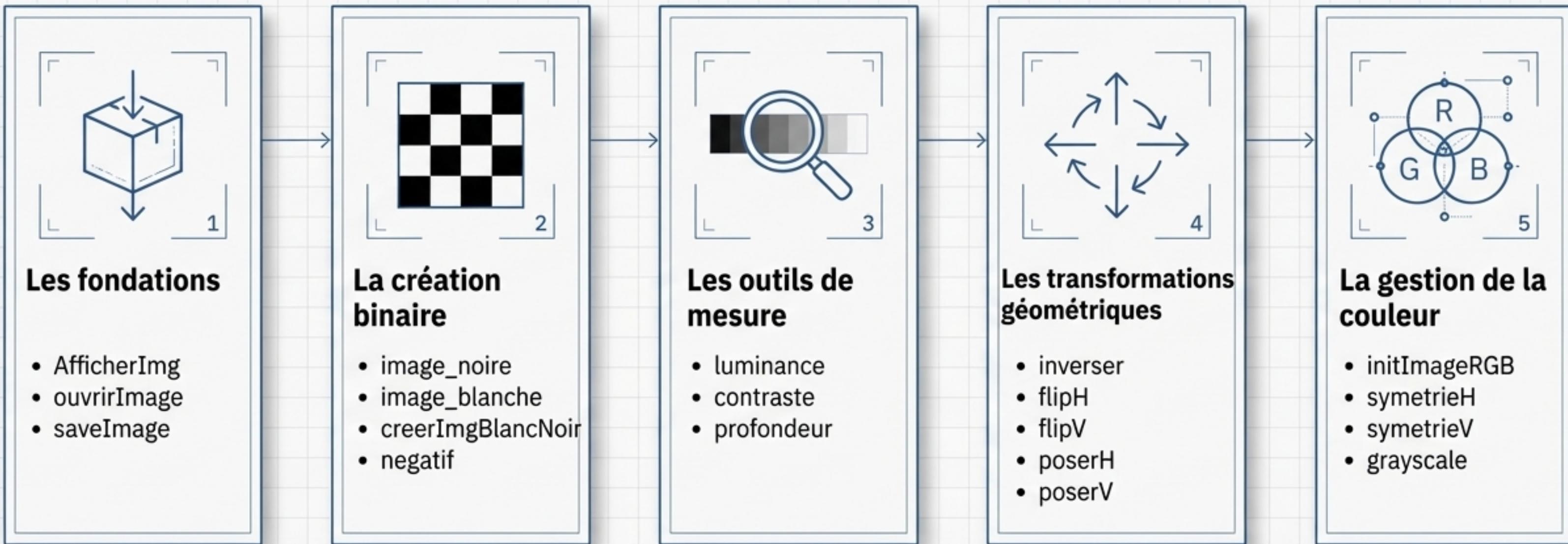


Algorithme

Pour chaque pixel, la nouvelle valeur de gris est la moyenne des composantes R, G, B minimale et maximale :
$$(\max(R,G,B) + \min(R,G,B)) / 2$$
.

Bilan de la Boîte à Outils : Une Bibliothèque Complète

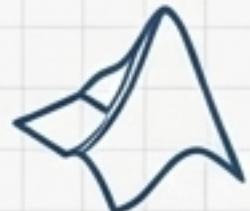
Le projet a été structuré en cinq ensembles de fonctions cohérentes, chacun remplissant un rôle précis dans le traitement des images.



Les objectifs fixés ont été atteints et les consignes du cahier des charges respectées, produisant un résultat satisfaisant.

Le Choix de l'Architecte : MATLAB vs. Python

Les deux environnements se sont révélés puissants et complémentaires. Le choix dépend essentiellement du contexte d'utilisation.



MATLAB

Forces

- Simplicité d'utilisation, environnement intégré, idéal pour le prototypage rapide et la manipulation matricielle. Fonctions natives efficaces.

Idéal pour

- L'enseignement, le prototypage scientifique.



Python (avec NumPy, Matplotlib)

Forces

- Grande flexibilité, richesse des bibliothèques (OpenCV), open-source, large communauté.

Idéal pour

- Le développement, la portabilité, l'intégration dans des projets de plus grande envergure.

Conclusion : L'utilisation conjointe de ces deux environnements permet de tirer profit des avantages de chacun et d'enrichir l'approche du traitement d'images.

Un Projet Collaboratif

Nous remercions notre encadrant pour son accompagnement, ainsi que toutes les personnes ayant contribué à la réussite de ce projet.

