

# Compte-Rendu - SAE

RA-IL2 : Chauvel Clément, Gesnel Kerrian, Micheli Thomas, Raouf Achraf

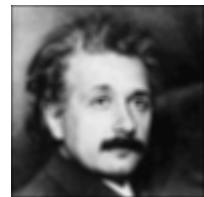
## Prétraitement de l'image

Dans le cadre du prétraitement de l'image, peu importe le filtre utilisé, l'algorithme était sensiblement similaire.

### 1. Flou par moyenne

Le travail est effectué en plusieurs étapes :

- Lecture de l'image : L'image est lue à partir du chemin d'accès fourni et une nouvelle image est créée avec les mêmes dimensions.
- Application du filtre : Pour chaque pixel de l'image, le filtre de flou moyen est appliqué. Le filtre est une matrice 3x3 où chaque élément est 1.0. Pour chaque pixel, les 9 pixels environnants (y compris le pixel lui-même) sont pris en compte. Si un pixel environnant est dans les limites de l'image, sa couleur est récupérée et multipliée par l'élément correspondant du filtre. Les résultats sont ensuite additionnés et divisés par 9 pour obtenir la nouvelle couleur du pixel.
- Vérification des limites de la couleur : Les valeurs de couleur obtenues sont limitées entre 0 et 255.
- Application de la couleur : La nouvelle couleur est appliquée au pixel correspondant dans la nouvelle image.
- Sauvegarde de l'image : La nouvelle image est enregistrée avec le suffixe "-flouMoyenne" ajouté au nom du fichier original. Le format de l'image (jpg, png, jpeg) est conservé.

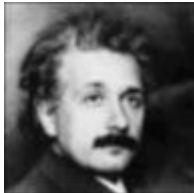


### 2. Flou gaussien

Le travail est effectué en plusieurs étapes :

- Lecture de l'image : L'image est lue à partir du chemin d'accès fourni et une nouvelle image est créée avec les mêmes dimensions.
- Définition du filtre : Un filtre gaussien est défini en fonction de la taille spécifiée. Les filtres de taille 3, 5 et 7 sont prédéfinis avec des coefficients correspondants.
- Application du filtre : Pour chaque pixel de l'image, le filtre gaussien est appliqué. Pour chaque pixel, les pixels environnants sont pris en compte en fonction de la taille du filtre. Si un pixel environnant est dans les limites de l'image, sa couleur est récupérée et multipliée par l'élément correspondant du filtre. Les résultats sont ensuite additionnés et divisés par le coefficient du filtre pour obtenir la nouvelle couleur du pixel.

- Vérification des limites de la couleur : Les valeurs de couleur obtenues sont limitées entre 0 et 255.
- Application de la couleur : La nouvelle couleur est appliquée au pixel correspondant dans la nouvelle image.
- Sauvegarde de l'image : La nouvelle image est enregistrée avec le préfixe "Gaussien-new-" ajouté au nom du fichier original. Le format de l'image (jpg) est conservé.



Filtre de taille 3



Filtre de taille 5



Filtre de taille 7

## Détection et visualisation des biomes

### 1. Détection des groupes de pixels de couleur similaire

Choix de l'algorithme de clustering le plus adapté à cette tâche. Prenez en compte notamment le nombre de pixels à traiter ?

Étant donné que la détection, le traitement et la visualisation se fait sur l'ensemble des pixels de l'image de la planète, il est nécessaire de faire un algorithme avec une complexité réduite, à savoir, KMeans.

Choix de la métrique pour calculer la similarité entre couleurs (voir TP de préparation).

Pour calculer la similarité entre les couleurs, nous avons opté pour la norme CIELAB qui permet d'avoir un rendu uniforme (la similarité d'un point de vue métrique et la même d'un point de vue visuel).

Choix des paramètres du clustering. Il est conseillé de tester différentes valeurs.

Dans le cadre de KMeans, le choix des paramètres est simple, il s'agit du nombre de clusters et le nombre d'itérations (les biomes donc 10 clusters et une centaine d'itérations).

Évaluation de la qualité du résultat obtenu. La validation peut être visuelle ou en utilisant des indices de qualité.

On a décidé de partir sur une validation visuelle qui semblait plutôt cohérente avec l'image de base au niveau des différents clusters.



Cette partie a nécessité deux classes et une interface, à savoir AlgoClustering, qui est une interface, KMean qui implémente l'interface et qui introduit l'algorithme vu en cours, et ImageBiomes qui utilise un objet KMean pour créer des images.

La classe ImageBiomes utilise l'algorithme KMeans pour regrouper les couleurs d'une image et les remplacer par la couleur la plus proche d'une palette prédéfinie. Voici un résumé des étapes :

- ImageBiomes utilise un objet KMeans et un objet Palette

- L'image est lue et ses couleurs sont extraites dans un tableau 2D ('descriptions'), où chaque ligne représente un pixel et les colonnes représentent les composantes rouges, verte et bleue de la couleur.
- L'algorithme `KMeans` est appliqué à ce tableau, regroupant les couleurs similaires. La méthode `cluster` de l'objet `KMeans` est appelée avec le tableau `descriptions` comme paramètre. Cette méthode renvoie un tableau ('clusters') où chaque élément est le numéro de cluster du pixel correspondant.
- Pour chaque cluster, la couleur du pixel correspondant dans l'image originale est récupérée et la couleur la plus proche de la palette est trouvée. Cette couleur est ensuite définie dans la liste `biomes` à l'index correspondant au cluster.
- Une nouvelle image ('newImg') est créée avec les mêmes dimensions que l'image originale. Chaque pixel de cette image est défini à la couleur de la liste `biomes` qui correspond au cluster du pixel dans l'image originale.
- Pour chaque couleur dans la liste `biomes`, une nouvelle image est créée où les pixels du cluster correspondant sont définis à la couleur et tous les autres pixels sont définis à une version plus claire de leur couleur originale. Ces images sont sauvegardées avec le nom de l'image originale, "-Biome", le nom du biome correspondant à la couleur, et l'extension de fichier appropriée.

## Détection et affichage des écosystèmes pour chaque biome

Choix de l'algorithme de clustering le plus adapté à cette tâche. Prenez en compte notamment le nombre de pixels à traiter ?

On a utilisé l'algorithme DBScan.

Choix de la métrique pour calculer la similarité entre couleurs (voir TP de préparation).

Pour calculer la similarité entre les couleurs, nous avons opté pour la norme CIELAB qui permet d'avoir un rendu uniforme (la similarité d'un point de vue métrique et la même d'un point de vue visuel).

Choix des paramètres du clustering. Il est conseillé de tester différentes valeurs.

Dans le cadre de DBScan il n'est pas nécessaire de choisir un nombre de cluster, par contre il faut un nombre de points minimum et un epsilon.

Évaluation de la qualité du résultat obtenu. La validation peut être visuelle ou en utilisant des indices de qualité.

On a décidé de partir sur une validation visuelle qui semblait plutôt cohérente avec l'image de base au niveau des différents clusters.



La classe ImageEcosystemes utilise l'algorithme DBSCAN pour regrouper les couleurs d'une image et les remplacer par la couleur la plus proche d'une palette prédéfinie. Voici un résumé des étapes :

- ImageEcosystemes utilise un objet DBSCAN et un objet Palette.
- L'image est lue et ses couleurs sont extraites dans un tableau 2D (descriptions), où chaque ligne représente un pixel et les colonnes représentent les composantes rouges, vertes et bleues de la couleur.
- L'algorithme KMeans est appliqué à ce tableau, regroupant les couleurs similaires. La méthode cluster de l'objet KMeans est appelée avec le tableau descriptions comme paramètre. Cette méthode renvoie un tableau (clusters) où chaque élément est le numéro de cluster du pixel correspondant.
- Pour chaque cluster, une nouvelle image (newImg) est créée avec les mêmes dimensions que l'image originale. Les pixels de ce cluster sont ensuite extraits et regroupés dans un nouveau tableau (clusterData).
- L'algorithme DBSCAN est appliqué à ce nouveau tableau, créant des sous-clusters. La méthode cluster de l'objet DBSCAN est appelée avec le tableau clusterData comme paramètre. Cette méthode renvoie un tableau (clusterColors) où chaque élément est le numéro de sous-cluster du pixel correspondant.
- Pour chaque sous-cluster, la couleur du pixel correspondant dans l'image originale est récupérée et la couleur la plus proche de la palette est trouvée. Cette couleur est ensuite définie dans la liste des biomes à l'index correspondant au sous-cluster.
- Chaque pixel de l'image newImg est défini à la couleur de la liste des biomes qui correspond au sous-cluster du pixel dans l'image originale. Si un pixel n'appartient pas au cluster actuel, sa couleur est définie à une version plus claire de sa couleur originale.
- L'image newImg est sauvegardée avec le nom de l'image originale, le numéro du cluster, "-DBSCAN", et l'extension de fichier appropriée.