

---

# TP #2 – CLUSTERING DE COULEURS

---

## 8INF954 – Forage de données

Dans le cadre du cours, nous avons vu l'utilisation de l'apprentissage machine, et notamment du clustering, avec l'utilisation de k-Means et DBScan. Afin d'appliquer les concepts vus en cours et d'avoir un point de vue pratique dans le cours, il est demandé d'implémenter une version de K-means et de DBScan dans le langage de votre choix.

***Ce TP peut être réalisé seul ou en binôme.***

### Travail à effectuer

Le but du TP est de concevoir un programme dans votre langage préféré. Le programme devra répondre aux exigences suivantes :

1. Prendre une image en paramètre (drag & drop ou input type « parcourir »)
2. Convertir l'image en pixels RGB. La gestion de la transparence (filtre alpha) n'est pas demandée ici.
3. La suite du TP se distingue en deux catégories.
  - 3.1. K-Means
    - Réaliser un clustering de couleurs en prenant en compte les trois dimensions (rouge, bleu, vert)
    - Tester plusieurs valeurs pour k, à savoir 16, 32 et 64.
    - Associer chaque couleur de pixel au centre de son cluster associé.
    - Remplacer l'ensemble des nouvelles valeurs dans un tableau et redessiner l'image à la même taille
  - 3.2. DBScan
    - Réaliser un clustering de couleurs, en prenant en compte les trois dimensions (rouge, bleu, vert)
    - Tester plusieurs valeurs de distance inter-classes
    - Associer chaque couleur de pixel au centre de son cluster associé
    - Remplacer l'ensemble des nouvelles valeurs dans un tableau et redessiner l'image à la même taille
4. A chaque test effectué, enregistrez l'image de fin, et observez les différences liées aux changements de paramètres. Donnez une explication selon vous sur l'implication des paramètres sur le résultat final.

5. Tester la différence entre une distance euclidienne et une distance de Manhattan. Indiquez les différences obtenues pour chaque algorithme.
6. *Un bonus sera accordé pour un TP mettant en place une interface graphique complète.*

## Format de rendu

Le projet DOIT respecter cette structure de dossiers.

- L'ensemble de votre implémentation doit être dans un **dossier « src »**, à la racine du zip.
- Un fichier **README.md** doit être présent à la racine du zip et explique comment lancer le fichier. Il est nécessaire que les instructions soient claires et fonctionnelles. Si le code ne peut pas être exécuter avec les instructions fournies dans le README, une grande pénalité sera appliquée ( $\approx 60\%$ )
- Les images que vous avez essayées devront être stockées dans un **dossier « img »** à la racine du zip
- Les images sorties de vos algorithmes doivent être dans un **dossier « output »** et chaque image doit avoir un nom représentatif de son origine.  
*Par exemple, « kmeans\_16.png », « kmeans\_32.png », ou encore « DBScan\_30.png ».*
- Un **fichier PDF** à la racine du zip répondant aux points 4 et 5 de ce TP.

**Les binômes devront avoir un rendu plus approfondi que ceux qui réaliseront le TP seuls.**

Pour être valide, il faudra rendre un UNIQUE fichier .zip (un seul par binôme également) comportant l'ensemble des documents énoncés ci-avant. Ce fichier devra respecter la nomenclature suivante : « TP1\_NOM\_PRENOM.zip » (« TP1\_NOM1\_PRENOM1\_NOM2\_PRENOM2.zip » pour un binôme) en remplaçant évidemment par vos noms et prénoms respectifs.

*Tout non-respect de ces consignes de rendu peut se voir sanctionné par des pénalités.*

## Date de rendu

Le projet devra être remis avec **un seul fichier .zip au plus tard le lundi 8 mars 2022 à 23h59**. Le rendu se fera sur Moodle via l'option créée à cet effet.

**ATTENTION : Aucun retard n'est toléré. Si vous (ou votre binôme) n'avez pas rendu le fichier zip, vous aurez automatiquement la note 0 (votre binôme également).**