

# TP1 : Traitement d'Image

## Overview Haut Niveau

Said Ohamouddou

### 1 Objectifs

Ce TP présente une approche complète du traitement d'image, des techniques classiques aux méthodes d'apprentissage automatique. Nous utiliserons le dataset **Leaf Images** de Kaggle : <https://www.kaggle.com/datasets/ichhadhari/leaf-images>

### 2 Partie 1 : Traitement d'Image Classique (OpenCV)

#### 2.1 Création d'Images

- Créer une image RGB depuis zéro
- Créer une image en niveaux de gris
- Créer une image binaire

#### 2.2 Analyse d'Images Réelles

- Charger et afficher une image du dataset
- Afficher les caractéristiques (dimensions, type, taille)
- Générer et interpréter l'histogramme des couleurs
- Normaliser l'image
- Modifier un ensemble de pixels

#### 2.3 Transformations

- Conversion couleur → niveaux de gris
- Seuillage et détection de contours
- Débruitage avec différents filtres
- Opérations morphologiques : dilatation et érosion
- Compression avec/sans perte

## **3 Partie 2 : Machine Learning (Scikit-Learn)**

### **3.1 Extraction de Caractéristiques**

Utiliser le dataset Leaf Images pour extraire :

- Descripteurs HOG (Histogram of Oriented Gradients)
- Descripteurs SIFT (Scale-Invariant Feature Transform)
- Motifs LBP (Local Binary Patterns)
- Histogrammes de couleurs

### **3.2 Classification**

Appliquer tous les algorithmes de classification de Scikit-Learn :

- SVM, Random Forest, k-NN
- Naive Bayes, Régression logistique
- Gradient Boosting

### **3.3 Évaluation**

- Comparer les performances des différents descripteurs
- Analyser les matrices de confusion
- Interpréter les résultats de classification

## **4 Partie 3 : Réseaux de Neurones Simples (MLP)**

### **4.1 Préparation**

- Aplatir les images du dataset Leaf Images
- Normaliser les données d'entrée
- Encoder les labels des classes

### **4.2 Modélisation**

- Créer un MLP avec Scikit-Learn, Keras ou PyTorch
- Définir l'architecture (couches cachées, activation)
- Entraîner le modèle

### **4.3 Interprétation**

- Évaluer les performances
- Comparer avec les méthodes classiques
- Analyser les avantages/inconvénients

## **5 Partie 4 : Réseaux de Neurones Convolutionnels (CNN)**

### **5.1 Architecture CNN**

Avec Keras ou PyTorch :

- Concevoir un CNN pour la classification des feuilles
- Utiliser des couches de convolution, pooling, dropout
- Configurer la couche de sortie selon le nombre de classes

### **5.2 Entraînement**

- Appliquer l'augmentation de données
- Optimiser les hyperparamètres
- Utiliser des callbacks (early stopping, learning rate)

### **5.3 Comparaison et Interprétation**

- Comparer CNN vs MLP vs méthodes classiques
- Analyser les performances sur le dataset
- Visualiser les filtres appris
- Interpréter les résultats finaux

## **6 Livrables**

1. Code source commenté pour chaque partie
2. Rapport d'analyse des résultats
3. Graphiques de comparaison des performances
4. Conclusions sur l'efficacité des différentes approches