**INF5081 Gestion et analyse de données – Hiver 2025**

**Travail pratique #2**

**Professeur : Ataky Tsham Mpinda, Steve**

**Nom : Andy Kouassi Code permanent :**

**Nom : Oswald Essongue Kumambuelet Code permanent : ESSO16019809**

# 

## 

## 

## 

## 

## **1. Énoncé**

**Titre du projet :** Système de Récupération d'Images Basé sur le Contenu (CBIR)

**Objectif :** Développer une application complète de recherche d’images similaires sans métadonnées textuelles, uniquement à partir de leur contenu visuel (CBIR), en s’appuyant sur des réseaux de neurones pré-entraînés et le clustering par K-Means.

## **2. Introduction**

La recherche d’images basée sur le contenu (CBIR) repose sur l’analyse des caractéristiques visuelles extraites automatiquement d’images. Les modèles de deep learning, notamment les réseaux de neurones convolutionnels (CNN), offrent la capacité de générer des descripteurs denses et discriminants permettant de comparer les images visuellement.

Ce projet exploite trois modèles pré-entraînés (VGG16, ResNet50 et InceptionV3) pour extraire des vecteurs de caractéristiques à partir d’un jeu d’images. L’algorithme K-Means est ensuite appliqué pour regrouper les images par similarité. Une interface utilisateur permet de téléverser une image requête, choisir une métrique de distance, et visualiser les images les plus proches accompagnées d’une analyse statistique de leur répartition.

## **3. Problématique**

Le projet pose plusieurs enjeux :

* **Extraction de caractéristiques** : Comment exploiter efficacement des CNN pré-entraînés pour produire des vecteurs représentatifs ?
* **Clustering** : Quel est le bon nombre de clusters pour structurer efficacement la base ?
* **Distance** : Quelle métrique de distance produit les résultats les plus cohérents visuellement ?
* **Interface** : Comment permettre une interaction fluide pour l’utilisateur et afficher les résultats de façon intuitive et significative ?

## **4. Développement**

### **4.1 Architecture globale du système**

#### **A. Phase Offline : Extraction, Clustering et Sauvegarde**

1. **Extraction des caractéristiques :**
   * Chargement et redimensionnement des images pour chaque modèle (224x224 pour VGG16/ResNet50, 299x299 pour InceptionV3).
   * Passage des images dans les modèles CNN pré-entraînés pour produire des vecteurs distincts.
2. **Clustering par K-Means :**
   * Clustering des vecteurs avec un k fixé (ex. : 5), ou optimalement choisi via la méthode du coude.
   * Attribution d’un label à chaque image selon son cluster.
3. **Sauvegarde :**
   * Enregistrement dans des fichiers .npy des vecteurs, des labels de cluster et des chemins d’images correspondants, par modèle.

#### **B. Phase Online : Interface Utilisateur**

1. **Interface Web (Streamlit) :**
   * Upload d’une image requête.
   * Choix du modèle de descripteur (VGG16, ResNet50, InceptionV3).
   * Choix d’une métrique de distance (Euclidienne, Manhattan, Chebyshev, Canberra).
2. **Traitement de l’image requête :**
   * Prétraitement selon le modèle choisi.
   * Extraction du vecteur de l’image.
3. **Calcul des similarités :**
   * Comparaison du vecteur de l’image requête aux vecteurs de la base offline.
   * Récupération et affichage des 5 images les plus proches (distance minimale).
4. **Analyse statistique :**
   * Comptage des labels des images similaires.
   * Génération d’un histogramme de fréquence des labels pour vérifier la cohérence du cluster.

### **4.2 Technologies et Outils**

* **Langage :** Python
* **Deep Learning :** TensorFlow/Keras
* **Clustering :** scikit-learn (KMeans)
* **Traitement d’images :** Pillow
* **Données :** NumPy, Pickle
* **Interface :** Streamlit
* **Visualisation :** Matplotlib

## **5. Analyse des Résultats**

Pour évaluer la performance des différents modèles, une série de tests a été menée sur les **5 premières images du dataset**. Chaque image a été soumise aux trois descripteurs de caractéristiques (VGG16, ResNet50, InceptionV3), en utilisant une **même métrique de distance (Euclidienne)**.

L’évaluation s’est basée sur la pertinence visuelle des images retournées et l’homogénéité des labels dans l’histogramme. Chaque requête a été notée sur une **échelle de 1 à 5** selon la cohérence des résultats :

* **5** : Toutes les images retournées sont du même type (ex. : chats 🐱) → *très pertinentes*
* **4** : 4 images sur 5 sont similaires → *pertinentes*
* *(...)*

Voici les résultats obtenus :

| Image Requete | VGG16 | ResNet50 | InceptionV3 |
| --- | --- | --- | --- |
| Image 1 | 3 | 4 | 5 |
| Image 2 | 5 | 5 | 5 |
| Image 3 | 2 | 3 | 3 |
| Image 4 | 4 | 1 | 4 |
| Image 5 | 1 | 1 | 4 |
| Moyenne | 3 | 2.8 | 3.6 |

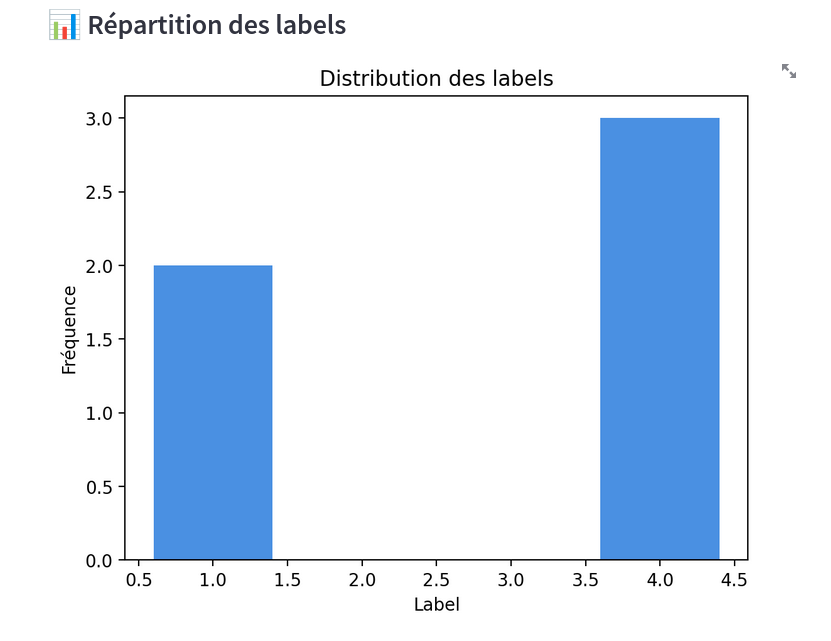
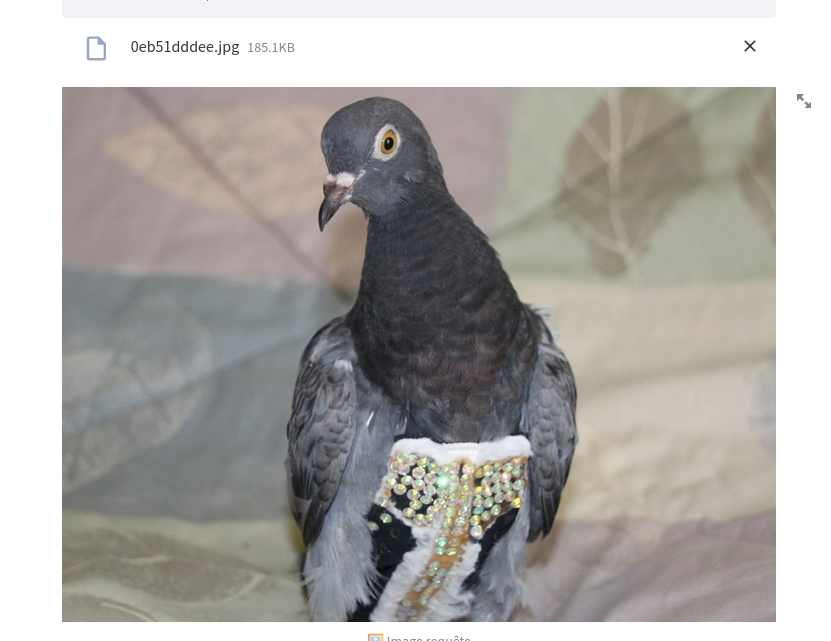
Le modèle **InceptionV3** s’est distingué avec la meilleure moyenne (**3.6**), confirmant ainsi sa capacité à capturer efficacement les caractéristiques visuelles dans ce contexte.

## **6. Discussion**

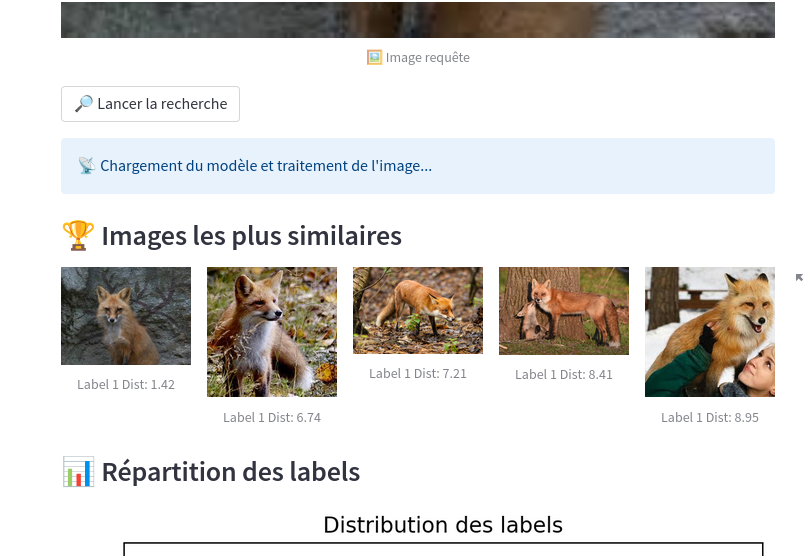
## **7. Analyse des clusters**

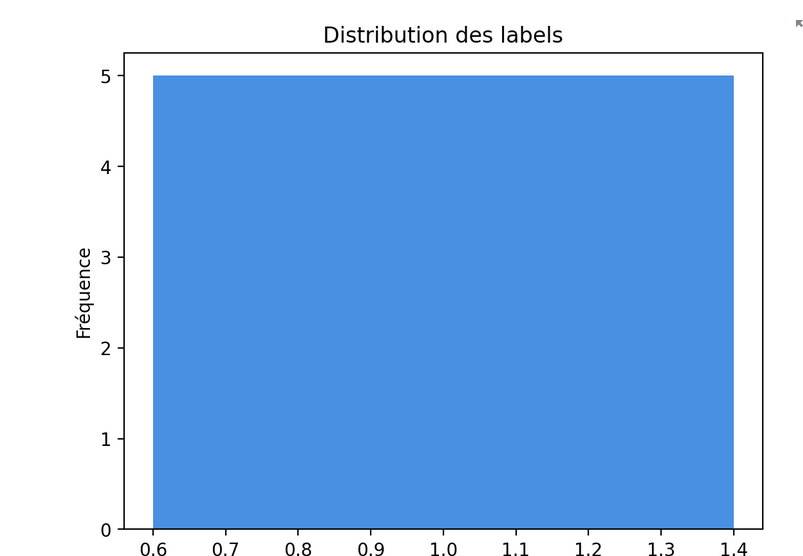
L’histogramme des labels montre que, pour certaines requêtes, les images similaires se trouvent majoritairement dans un même cluster, ce qui témoigne d’un bon regroupement visuel effectué par K-Means. Cependant, pour d'autres requêtes, les images similaires sont issues de clusters variés, ce qui pourrait indiquer une certaine confusion dans la représentation des caractéristiques visuelles ou un manque d’optimisation du nombre de clusters.

Voici un exemple avec 2 clusters:



En voici un avec le même cluster:





## **8. Conclusion**