Enoncé du projet « ML & DL for Multimedia Retrieval »

I. Introduction :

Comme annoncé en séance de cours, ce projet s'appuiera sur les connaissances acquises durant le cours (Fig.1) pour développer une application d'indexation et recherche multimédia.

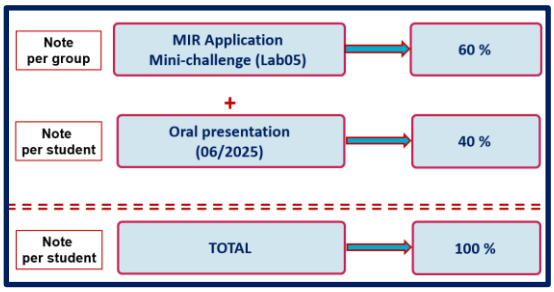

Figure 1: Modalités d'évaluation de l'AA

Les projets seront réalisés par groupe de deux avec les échéances suivantes :

- Date de remise du projet (rapport d'environ 20 pages + code + manuel) : le **16/06/2025** via Moodle
- Mode de présentation du projet : en présentiel (**le jour de l'examen en session**)
- Date de présentation du projet : **23/06/2025** ou chaque groupe aura **15 minutes** au maximum pour présenter son projet suivi de **5 à 10 minutes** de questions.

L'ordre de passage sera :

- **08h30 – 08h55 :** Groupe 01 (Cyril Moreau et Aymeline Giloteau)
- **08h55 – 09h20 :** Groupe 02 (Muscato Aysel et Robette Marion)
- **09h20 – 09h45 :** Groupe 03 (Clément Allard et Jordan Demaret)
- **09h45 – 10h10 :** Groupe 04 (Hugo DARVAI et Hao YE)

********** **PAUSE** **********

- **10h25 – 10h50 :** Groupe 05 (Mathieu Zilli et Azzizi Mohssine)
- **10h50 – 11h15 :** Groupe 06 (Nicolas Melaerts et Kenza Khemar)
- **11h15 – 11h45 :** Groupe 07 (Liénard Benjamin et Vanderberken Medhi)
- **11h40 – 12h05 :** Groupe 08 (Bourez Elisa et RUIZ AYA Johan)
- **12h05 – 12h30 :** Groupe 09 (Cyril Taquet et Windal Thibaut)
- **13h30 – 13h55 :** Groupe 10 (FARIS Mourad et MENAS TAMASI Rafael)
- **13h55 – 14h15 :** Groupe 11 (Teubou Melonou Jospin)
- **14h15 – 14h35 :** Groupe 12 (Radelet Alexandre)

Note 1 : lors de la présentation de vos projets, vous aurez aussi des questions liées à la théorie vue au cours.

II. Partie I : Enoncé du projet « moteur de recherche d'images »

Le but du projet est de développer un moteur de recherche exploitant # descripteurs, il faudra :

1. Indexer la base de données avec les descripteurs de votre choix. Si plusieurs descripteurs sont choisis, il faudra donner la possibilité de les combiner (**voir Enoncé TP3.3 : fortement conseillé**) ;
2. Réaliser la recherche en donnant la possibilité de choisir la fonction de calcul de similarité (Euclidéenne, Corrélation, Chi-square, Bhattcharyya, Brute Force Matcher, Flann, etc.) ;
3. Afficher le Top20 et Top50 pour les images requêtes ;
4. Calculer le Rappel (R), Précision (P), Average Precision (AP), Mean Average Precision (MaP) et R-Precision

Vous avez le choix de travailler sous Python ou C++ mais ce choix devra être pris en compte dans la partie facultative qui consiste à héberger votre application sur ressource Cloud.

- Les groupes **1, 3, 5 et 7, 9 et 11** travailleront sur la base de données « **Animaux** » contenant 05 classes dont chacune contient **06** classes de races d'animaux (**30 classes**). Pour tester le moteur, il faudra faire les requêtes suivantes **Lien de la BD :** https://github.com/sidimahmoudi/facenet_tf2/releases/download/AI_MIR_CLOUD/MIR_DATASETS_B.zip

Indice requête	Classe	Images
R1, R2, R3	**araignées**	0_5_araignees_tarantula_795, 0_4_araignees_gardenspider_631, 0_1_araignees_wolfspider_259
R4, R5, R6	**Chiens**	1_0_chiens_Siberianhusky_849, 1_3_chiens_Chihuahua_1315, 1_1_chiens_Labradorretriever_1054
R7, R8, R9	**Oiseaux**	2_2_oiseaux_greatgreyowl_2092, 2_4_oiseaux_robin_2359, 2_3_oiseaux_bluejay_2232

- Les groupes **2, 4, 6, 8, 10** et **12** travailleront sur la même base de données « **Animaux** » contenant **05** classes dont chacune contient **06** classes de races d'animaux (**30 classes**). Pour tester le moteur, il faudra faire ces requêtes :

Indice requête	Classe	Images
R1, R2, R3	**Poissons**	3_4_poissons_eagleray_3310, 3_5_poissons_hammerhead_3495, 3_3_poissons_tigershark_3244
R4, R5, R6	**Chiens**	1_2_chiens_boxer_1146, 1_4_chiens_goldenretriever_1423, 1_5_chiens_Rottweiler_1578
R7, R8, R9	**Singes**	4_3_singes_squirrelmonkey_4082, 4_2_singes_gorilla_4004, 4_1_singes_chimpanzee_3772

Les résultats de calcul des descripteurs « indexation » devront se présenter selon le tableau 1 ci-dessous :

Tableau 1: Mesures de performances d'indexation et recherche

Vos meilleurs descripteurs	Nom de(s) descripteur(s)	Temps d'indexation (s)	Taille du descripteur (MB)	Temps de recherche moyen par image (s)
Descripteur N° 01				
Descripteur N° 02				
Descripteur N° 03				

Les résultats attendus pour chaque requête devront se présenter comme suit :

Tableau 2: Mesures de précision du moteur recherche

Indice requête	R		P		AP		MaP	
	Top50	Top100	Top50	Top100	Top50	Top100	Top50	Top100
R1								
R2								
…								
R9								

Vous pouvez également ajouter une colonne TopMax (Max : nombre d'image par la classe concernée).

Note 2 : en raison de la complexité de son calcul, vous pourrez réduire la résolution des images pour calculer SIFT.

• II. Partie 02 « moteur de recherche multimodal »

Dans cette partie, nous allons exploiter le potentiel qu'offre la **recherche multimodale**, c'est à dire l'utilisation de plusieurs types de données pour rechercher. Dans notre cas, nous allons utiliser les images et textes. Nous vous conseillons de suivre ces étapes :

1. **Extraction des caractéristiques pour la partie vision :** vous pouvez réutiliser la partie n°1 mais il est demandé d'ajouter au minimum un vision transformer si cela n'a pas déjà été fait.
2. **Extraction des caractéristiques pour la partie texte** : les descriptions des images sont stockées sous forme de dictionnaire dans ce fichier. Les clés du dictionnaire étant les noms des images correspondantes aux descriptions. Il est conseillé d'utiliser un modèle transformer du type mini-LM par exemple sans le réentraîner.
3. **Combinaison des caractéristiques** : il est demandé de combiner ces caractéristiques pour avoir une approche multimodale. Il existe plusieurs types de combinaisons possibles : addition, multiplication ou moyenne par exemple.

Le résultat final devra permettra une recherche uniquement sur l'image, uniquement sur le texte ou la combinaison des 2 (image et texte). Il faudra pouvoir analyser le rappel et la précision pour les images (uniquement image et multimodale) et pour le texte, il faudra apprécier le score de similarité entre la requête textuelle et la description de l'image (**pas de calcul de rappel et précision concernant la recherche par texte**).

Concernant l'indexation, vous pouvez donc générer les trois descripteurs : un pour les images séparément (**votre meilleur descripteur de la partie 1**), un pour les textes séparément et un dernier en fusionnant les deux. La fusion des deux descripteurs (image et texte) peuvent être faite soit via la concaténation ou la moyenne (après avoir normalisé la taille des deux descripteurs)

Pour la recherche, vous avez trois possibilités :

- Une image de requête pour avoir les images proches avec leurs textes : descripteurs « image » (Fig 2.a)
- Un texte de requête pour avoir les textes proches avec leurs images : descripteurs « texte » (Fig 2.b)
- Une image + texte pour avoir les images et textes proches : descripteurs fusionnés. (Fig 2.c)

Vous n'êtes pas obligés de tous expérimenter, une possibilité ou deux suffiront.

Les résultats peuvent être présentés à l'aide des métriques de rappel et précision concernant les images et scores de similarité concernant le texte (**pas besoin de fournir les courbes de rappel et précision pour la partie 2**).

- **Note :** les scores de **R** et **P** doivent être calculés par rapport au nombre d'images par classe **TopMAX.**

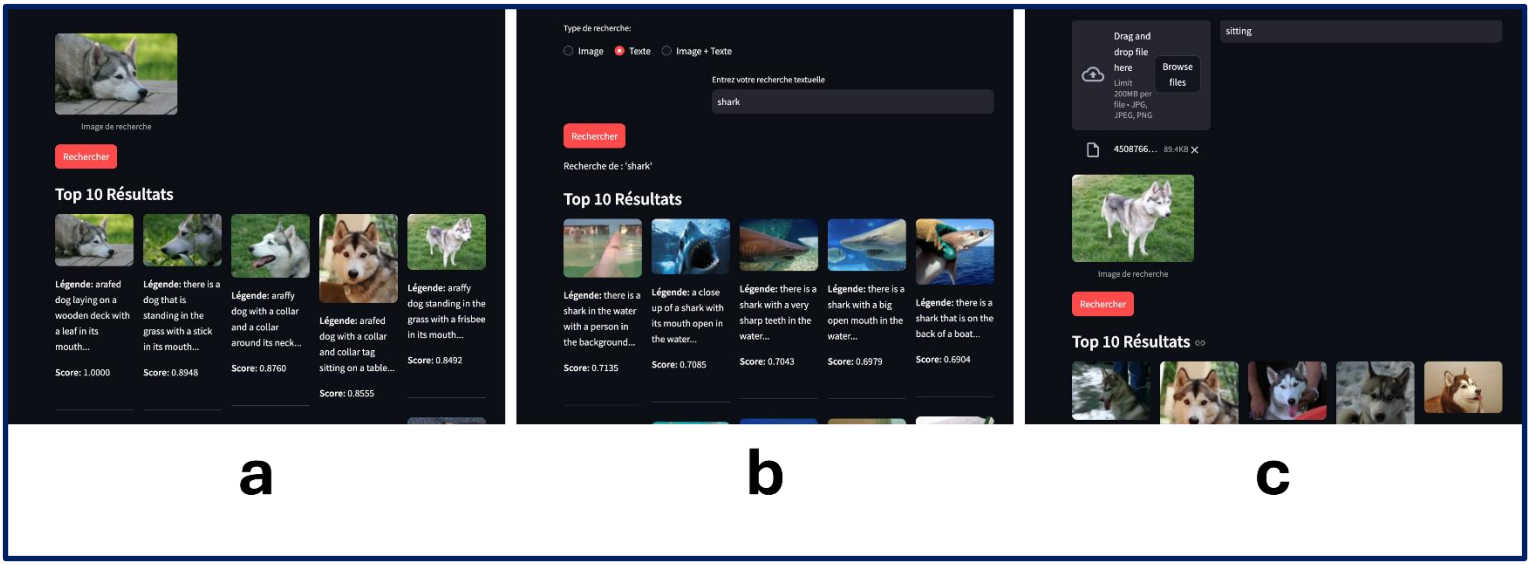

Figure 2: différents modes application de recherche multimodale

• III. Partie 03 « Architectures multimodale pour la recherche multimédia » : facultatif

Dans la partie 2, nous avons utilisé des modèles distincts pour extraire les caractéristiques des images et du texte. Cette approche permet une certaine flexibilité mais présente également des limitations, notamment en termes de fusion des informations. Une alternative consiste à utiliser un modèle pré-entraîné sur des paires image-texte, tel que **CLIP** (Contrastive Language-Image Pretraining) développé par OpenAI. Ce type de modèle est spécifiquement conçu pour apprendre une représentation commune entre images et descriptions textuelles. Les avantages de cette approche sont les suivants :

- **Représentation commune** : Un même espace vectoriel est utilisé pour les images et le texte, ce qui améliore la recherche croisée (retrouver une image à partir d'un texte ou un texte à partir de l'image).
- **Alignement naturel des modalités** : Les modèles comme **CLIP** sont entraînés sur de larges bases de données d'images et de descriptions, permettant une meilleure correspondance sémantique entre le texte et l'image.
- **Meilleure généralisation** : Contrairement aux approches utilisant des modèles distincts, un modèle multimodal peut mieux s'adapter aux nouvelles données et aux requêtes textuelles complexes.

Les étapes à suivre pour faire cette partie sont les suivantes :

1. **Télécharger et travailler sur une autre BD multimodale : https://paperswithcode.com/dataset/flickr30k**
2. **Utilisation d'un modèle multimodal pour indexer les images et le texte**
   a. **Indexation des images** : Transformer chaque image de la BD en un vecteur d'embedding avec CLIP.
   b. **Indexation des descriptions** : Passer chaque description textuelle dans CLIP pour obtenir un vecteur
3. **Stockage et indexation**
   a. Construire une base de données d'embeddings contenant les représentations des images et textes.
   b. Utiliser un moteur d'indexation. Il est conseillé d'utiliser **FAISS** pour la recherche rapide des voisins.
4. **Recherche et évaluation**
   a. Implémenter une requête textuelle et rechercher les images les plus similaires dans l'espace latent.
   b. Tester également la recherche inverse : retrouver du texte à partir d'une image.
   c. Évaluer la qualité des résultats avec des métriques telles que **Précision, Rappel et MAP**.
5. **Comparaison avec la Partie 2 (optionnel)**
   a. Comparer théoriquement la performance de CLIP avec l'approche précédente (modèles séparés).

Note : la partie 3 sera réalisée sur une BD différentes de celles vues dans les parties 1 et 2.

• III. Partie 04 « Hébergement sur ressource Cloud » : facultatif

L'objectif de cette partie est d'héberger votre application de recherche multimédia (**de la partie 1**) sur une ressource Cloud ou Edge afin d'offrir un service sous forme de Software As A Service « **SAAS** ». Nous vous proposons de suivre ces six (06) étapes :

1. **Indexation « extraction de caractéristiques » en local :** en raison des performances limitées de votre machine virtuelle (pas de GPU), nous vous proposons de sélectionner votre meilleur modèle (s) et fichier de caractéristiques d'images avant de les copier vers votre machine virtuelle. La phase d'indexation ne doit donc pas être hébergée sur ressource cloud.
2. **Test et configuration de votre application de recherche sur ressource Cloud :** ici, il faudra installer et configurer votre machine virtuelle afin de tester votre application (**partie 1**) sur la ressource Cloud.
3. **Génération de l'image Docker regroupant les fonctionnalités de votre application :** ici, il faudra créer un Dockerfile regroupant les instructions nécessaires pour faire fonctionner votre application. Notons que votre image devra gérer :
    a. **En entrée :** une image requête ;
    b. **En sortie :** les indices des images les plus similaires + la courbe de Rappel/Précision.
4. **Développement d'une page Web pour faciliter l'accès au service SAAS :** ici, il faudra développer une page Web (avec flask ou django voire php) permettant de :
    a. Afficher les informations des développeurs du projet & description/fonctionnalités de votre application ;
    b. Lancer l'application de la recherche à l'aide de boutons, labels, etc.
    c. Afficher les résultats de la recherche : images similaires (avec taux de similarité) + courbes R/P
    d. Compléter les tableau 1 et tableau 2 (à inclure dans votre rapport)
5. **Configuration d'accès :** configurer l'accès à votre service à l'aide de votre @ IP et numéro de port au choix
6. **Personnaliser votre site :** selon votre imagination en incluant une page de connexion
7. **Réduction de dimensionalité :** réduire la taille de vos descripteurs sans perdre en précision du moteur
8. **Cybersécurité :** analyser votre site en termes de sécurité avant de l'améliorer dans ce sens

- **Note 3 :** pour la **partie 1**, vous avec le choix entre utiliser vos PC, Google Colab ou demander l'accès au cluster IG (un accès par groupe) ;
- **Note 4 :** pour la **partie 2**, on pourra augmenter les capacités de mémoire (stockage et RAM) et de calcul selon vos besoins. Ceci vous permettra d'installer tous les outils nécessaires.

La figure 2 illustre un exemple d'hébergement de l'application de recherche d'images en utilisant une image Docker et une page Web développée à l'aide de php et html. Vous pouvez également visualiser cette vidéo pour avoir une idée simple et claire du travail attendu.

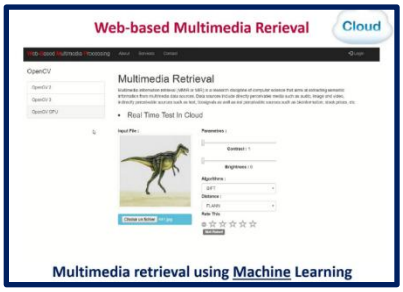

Figure 3: exemple d'hébergement d'application de recherche multimédia

IV. Quelques liens intéressants :

- Exemple d'hébergement d'une application **python** de classification d'images « Deep Learning » avec Docker et php : voir ce lien.
- Exemple d'hébergement d'une application **C++** de traitement d'images avec Docker et php : voir ce lien.

V. Contact: Sidi Ahmed Mahmoudi, Aurélie Cools et Maxime Gloesener