

## AA: Cloud & Edge Computing

### Enoncé du projet

#### I. Introduction :

Comme annoncé en séance de cours, ce projet s'appuiera sur les connaissances acquises durant les séances de cours et TP du module « Cloud & Edge Computing »

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 **17/06/2025** via Moodle
- Mode de présentation du projet : en présentiel
- Date de présentation du projet : le **24/06/2025** ou chaque groupe aura **13 minutes** au maximum pour présenter son projet suivi de **7 minutes** de questions.

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

L'ordre de passage :

- **13h30 – 13h50** : Groupe 01 « Roméo Ibraimovski et Maxime Dupuis »
- **13h50 – 14h10** : Groupe 02 « Jérémy Delnatte et Djoufo T. Tonesko »
- **14h10 – 14h30** : Groupe 03 « Gilles Jaunart et Pignozzi Agbenda »
- **14h30 – 14h50** : Groupe 04 « Mamadou Mounir Barry et Ouïam Bouazzati »
- **14h50 – 15h10** : Groupe 05 « Jordan Demaret et Xavier Delabie »

\*\*\*\*\* **PAUSE** \*\*\*\*\*

- **15h20 – 15h40** : Groupe 06 « Adrien Debuysschere et Adrien Zianne »
- **15h40 – 16h00** : Groupe 07 « Arthur Bongiovanni et Noah Minne »
- **16h00 – 16h20** : Groupe 08 « Duarte Santos et OMELYANENKO Mark »
- **16h20 – 16h40** : Groupe 09 « Maxim Guessar et Simon Moutier »
- **16h40 – 17h00** : Groupe 10 « Vancauwenberghe Emilien et Mohssine Azzizi »

\*\*\*\*\* **PAUSE** \*\*\*\*\*

- **17h10 – 17h30** : Groupe 11 « Clément Allard et Simon Michel »
- **17h30 – 17h50** : Groupe 12 « Agourzam Abdelhadi et El-Ismaïly Mohammed »
- **17h50 – 18h10** : Groupe 13 « MULLER Bastien et Lempire Rémy »
- **18h10 – 18h30** : Groupe 14 « KHEMAR Kenza et Melaerts Nicolas »

## II. Enoncé du projet :

Le but du projet est d'héberger une application d'indexation et recherche multimédia sur ressources Cloud (ou Edge). Nous vous proposons d'utiliser vos machines virtuelles créées dans le cadre du cours de « Cloud and Edge Computing » pour héberger vos applications. Le projet comprend deux parties :

### II.1. Partie 01 : indexation/recherche multimédia avec CNN et Vision Transformers

L'objectif de cette partie 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 ;
2. Réaliser la recherche en donnant la possibilité de choisir la fonction de calcul de similarité (Euclidéenne, Corrélation, Chi-square, Bhattacharyya, Brute Force Matcher, Flann, etc.) ;
3. Afficher le Top20 et Top50 pour les images requêtes ;
4. Calculer le Rappel (R), Précision (P) et courbes de rappel/précision.

**Facultatif** : calculer les métriques : Average Precision (AP), Mean Average Precision (MaP) et R-Precision.

Le code Python et solution du moteur de recherche (utilisant des modèles Deep Learning **CNN et Vision Transformers** développés via PyTorch) **est partagé via Moodle**. Votre travail consistera à héberger ce moteur de recherche. Il faudra tester et bien analyser l'application afin d'identifier le meilleur moyen de la déployer.

### II.2. Partie 02 : hébergement de l'application sur ressource Cloud ou Edge :

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, 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
5. **Configuration d'accès** : configurer l'accès à votre service à l'aide de votre @et numéro de port au choix
6. **Personnaliser votre site** : selon votre imagination en incluant une page de connexion
7. **Flexibilité d'utilisation** : permettre à l'utilisateur de choisir ou combiner entre les méthodes d'indexation et recherche (indexation à l'aide de 3 modèles au maximum. Par exemple : MobileNet, ResNet50 et Vit\_16)
8. **Combinaison de services** : docker-compose pour combiner les services (ajouter un service de votre choix)
9. **Utilisation de volumes Docker** : pour garder une trace des vos résultats après l'arrêt des conteneurs
10. **Cybersécurité** : analyser votre site en termes de sécurité avant de l'améliorer dans ce sens
11. **Mise à l'échelle** : programmer la mise à l'échelle en fonction de la charge de votre moteur (**TP4 facultatif**).

La figure 1 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 1: exemple d'hébergement d'application de recherche multimédia

### III. Quelques liens intéressants :

- Exemple d'hébergement d'une application **C++** de traitement d'images avec Docker et php : voir ce [lien](#).
- Exemple d'hébergement d'une app **python** de classification d'images « DNN » avec Docker et php : [lien](#).

### IV. Séances Projet : réservée à l'horaire du cours le 12/05 de 13h30 à 15h30.

### V. Contact: Sidi Ahmed Mahmoudi et Aurélie Cools