1 Consignes

⇒ **Lancez le TP sur Google Colab via l'adresse indiquée en début de séance.**

Le langage utilisé est Python 3 (+ pytorch).

Le code se trouve dans un notebook : `tp_classif_images.ipynb`. Ce code fait office de point de départ du TP. L'exécution de toutes les cellules débute l'entraînement d'un réseau de neurones profond pour une tâche de classification d'images entre deux catégories, sur 1 epoch (cf. ligne '`num_epochs = 1`' qui sera à modifier).

Les données sont stockées dans les dossiers `train/` (pour les données d'entraînement) et `test/` (pour les données de test). Il s'agit d'images provenant de la base de données "Cross-Age Celebrity Dataset" (CACD). La première catégorie (label '0') contient des images de visages de célébrités ayant un âge compris entre 20 et 26 ans, et la seconde catégorie (label '1') contient des images de visages de célébrités ayant un âge compris entre 52 et 62 ans. L'objectif de l'entraînement est de devenir capable de discriminer entre ces deux classes, c'est-à-dire de prédire si une image appartient à l'une ou à l'autre. Pour chacune des deux catégories, il y a 20,000 images dans l'ensemble d'entraînement, et 4,000 images dans l'ensemble de test. Toutes les images sont de taille 100x100 (pixels).

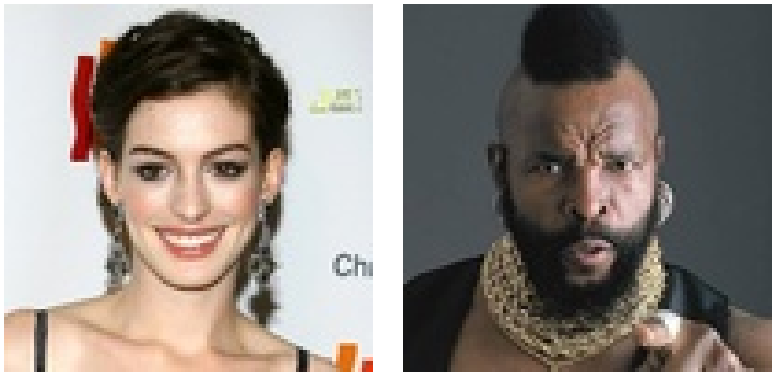

Figure 1: Deux images des 2 classes à discriminer. À gauche : label 0 (20∼26 ans), et à droite : label 1 (52∼62 ans).

⇒ **Un compte-rendu est à rendre à la fin des 4 heures de TP ou au plus tard une semaine après le TP (notations distinctes dans ces deux cas de figures).**

2 Questions

1) Décrivez la structure du réseau de neurones utilisé pour l'entraînement. Quel est le nombre total de paramètres de ce réseau ? Détailler la réponse.

2) Montrez que le réseau initialisé ne classifie pas mieux qu'un choix aléatoire.

3) Préparez le code de façon à enregistrer une courbe d'apprentissage pendant l'entraînement (c'est-à-dire la progression de la performance en fonction du nombre d'itérations). La courbe sera à insérer dans le compte-rendu.

4) Effectuez l'entraînement du réseau (viser une performance supérieure à 75 %, qui pourra être améliorée continuellement pendant les 4 heures du TP).

5) Évaluez les résultats obtenus avec des mesures plus précises que la simple performance. Pour une classe donnée, on pourra par exemple mesurer la moyenne et la variance des différences entre les estimation de probabilité de chaque label. Illustrez également vos résultats.

6) Donnez une liste d'hyperparamètres qui pourraient avoir un impact important sur l'entraînement, et expliquez leurs effets.

7) À quoi sert l'utilisation de la fonction `transforms.RandomCrop()` ? Remarque : on peut observer les transformations appliquées aux images en utilisant `original=False` dans les appels à la fonction `plotdata()`.

8) Mettez au point une procédure où plusieurs "crops" aléatoires sont utilisés pour évaluer la classe d'une image de visage. Vérifiez si cela améliore les performances de classification.

9) Cherchez des images de l'ensemble de données sur lesquelles le réseau entraîné se trompe complètement. Expliquez la procédure mise en place. Cherchez également le contraire : des images sur lesquelles le réseau se trompe le moins.

10) Généralisation hors de la base de données d'entraînement et de test : collectez (sur internet par exemple) une vingtaine d'images de visages centrées et ajustées de manière similaire à celles de la base de donnée (attention il suffit que l'image soit carrée), et testez le réseau entraîné pour vérifier qu'il fonctionne correctement.

11) Après vous être renseigné·e·s sur la méthode appelée "CycleGAN", décrivez comment cette dernière pourrait être appliquée afin d'entraîner un réseau à artificiellement vieillir ou rajeunir une image de visage.

12) (*Question subsidiaire*) Optimisation des entrées : comment feriez-vous pour construire une image d'entrée qui maximise le résultat de la classification par le réseau (par exemple en maximisant la probabilité estimée que le label soit 0) ? Si vous avez le temps, essayez de mettre cette méthode en oeuvre.