

TP 9 – Réseaux de neurones **(Teachable Machine)**

NOM : GHODBANI

PRÉNOM : Fares

Groupe : Miage

Réalisé avec Canva, Teachable Machine et Visual studio code.

1. Expérimentation

- **Création d'un modèle de classification d'images avec Teachable Machine.**
- **Classes initiales : chat et chien.**
- **Images utilisées : environ 10 images par classe pour l'entraînement et quelques images pour les tests.**

2. Observations sur le modèle initial

1. **Lorsqu'on teste une image hors des classes connues (ex : oiseau, personne), le modèle force une prédiction parmi les classes existantes :**
 - **Exemple : une image d'une personne → chat 100%.**
2. **Cela illustre que le modèle ne peut pas dire "je ne sais pas" et classe toujours selon la proximité avec les caractéristiques apprises.**

3. Ajout d'une nouvelle classe "rappeur"

- **Une nouvelle classe rappeur a été ajoutée avec des images de Drake.**
- **Tests :**
 - **Même photo que celle utilisée pour l'entraînement → 100% rappeur.**
 - **Photo similaire mais légèrement différente → confiance plus faible (ex : 85% rappeur, 15% autre).**
 - **Photo très différente ("goofy") → confiance encore plus faible.**

Conclusion : La confiance du modèle reflète la proximité avec les images d'entraînement : plus l'image est similaire, plus la confiance est élevée.

4. Tests croisés

- **Snoop Dog avec un chat → 67% chat, 33% rappeur.**
- **Drake avec un chien → 99% chien.**

Interprétation :

- **Le modèle attribue toujours la classe la plus proche, même si l'image appartient à une classe complètement différente.**
- **Cela illustre les limites des modèles simples de classification : ils ne comprennent pas "l'erreur" et ne détectent pas les images inconnues.**

Tous les tests effectués sont contenus dans les fichiers *test.jpeg, utilisés pour valider le modèle et observer son comportement sur différentes images.

5. Codes fournis

- **TM.js :**
 - **Contient le code JavaScript / P5.js et ml5.js pour utiliser le modèle Teachable Machine directement dans un navigateur.**
 - **Permet de classer les images en temps réel via la webcam.**
- **TM.py :**
 - **Contient un code Python / Keras qui fait exactement la même chose que le code JavaScript, mais en Python.**
 - **Il est basé sur le modèle exporté depuis Teachable Machine en format Keras (.h5).**
 - **Permet de classer des images locales ou d'adapter le code pour la webcam en Python.**
 - **Ce code est "convertissable" car Teachable Machine permet d'exporter le modèle pour différents langages : JavaScript pour le web, Python pour local ou serveur.**

Remarque :

- **La version Python est pratique si l'on veut tester le modèle hors navigateur, intégrer dans des scripts ou des applications Python, ou faire des expérimentations plus avancées (ex : traitement d'images en batch, intégration dans une IA locale).**

6. Conclusion générale

- **La Teachable Machine permet de créer rapidement des modèles de classification d'images simples.**
- **Les modèles fonctionnent bien pour des images proches de l'entraînement, mais sont limités sur des images inconnues ou ambiguës.**
- **L'ajout d'une classe "autres" peut aider à gérer les images hors des classes connues.**