

Projet : Image classification à l'aide du Deep Learning

Mise en situation

Vous êtes bénévole pour l'association de protection des animaux de votre quartier. C'est d'ailleurs que vous avez ainsi trouvé votre compagnon idéal, *Snooky*. Vous vous demandez donc ce que vous pouvez faire en retour pour aider l'association.



Votre première rencontre avec Snooky

Vous apprenez, en discutant avec un bénévole, que leur DB de pensionnaires commence à s'agrandir et qu'ils n'ont pas toujours le temps de référencer les images des animaux qu'ils ont accumulées depuis plusieurs années. **Ils aimeraient donc obtenir un algorithme capable de classer les images en fonction de la race du chien.**

Les données

Les bénévoles de l'association n'ont pas eu le temps de réunir les différentes images des pensionnaires dispersées sur leurs disques durs. Pas de problème, vous entraînerez votre algorithme en utilisant le [Stanford Dogs Dataset](#).

Vos missions

Vous avez peu d'expérience sur le sujet, vous décidez donc de contacter un ami expert en Deep Learning ; classification d'images.

Il vous conseille dans un premier temps de **pré-traiter des images avec des techniques spécifiques** (e.g. éventuellement modification de la taille des images, cropping, ...) et de réaliser de la **data augmentation**.

Ensuite, il vous incite à **mettre en œuvre deux approches** que vous comparerez en termes de temps de traitement et de résultat :

1. Une première **en réalisant votre propre réseau CNN** en prenant soin d'optimiser certains hyperparamètres. Vous pouvez vous inspirer de réseaux CNN existants.
2. Une deuxième **en utilisant le transfer learning**, c'est-à-dire en utilisant un réseau déjà entraîné, et en le modifiant pour répondre à votre problème.

Livrables attendus

1. Une organisation [DevOps Azure](#) qui contient :
 - Un **board** où vous répartissiez les tâches entre les membres de votre groupe suivant votre méthode de travail [Agile](#), [Scrum](#)
 - Les **pipelines** pour le CI/CD
2. Un repo [Github](#) où on trouve :
 - Un fichier README (où vous expliquez comment lancer les scripts, ...),
 - Un notebook Python (non cleané, pour comprendre votre démarche),
 - Le code du front où l'utilisateur peut uploader une image en entrée et le modèle retourne le Top 3 des races les plus probables du chien
 - Un support de présentation (environ 10 slides) :
 - Des principaux résultats (à votre manager) : *Accélération du travail d'indexation de la DB suite à la mise en place de l'algorithme de détection de la race du chien sur une photo*
 - Les explications au bénévole qui gère la DB pour lui montrer comment déployer le CNN dans l'application et la mettre en production.

Modalités de présentation du travail

Votre présentation pourra prendre cette forme (à titre indicatif) :

- | | |
|--------|---|
| 5 min. | Rappel de la problématique, présentation du jeu de données, de l'exploration, du cleaning, data augmentation |
| 15 min | Explication de l'approche pour mettre en place votre solution (méthodo, code, interprétation des résultats, modèle final sélectionné ainsi que des performances et améliorations effectuées, etc ?) |
| 10 min | Séance de questions-réponses |

Ressources complémentaires

- Guide d'utilisation de [Google Colaboratory avec TPU](#)
- Profiter de 300\$ pour entraîner vos modèles avec [GCP VertexAI](#)
- Ce [tutoriel](#) présente les concepts de data augmentation. À vous de voir si cela peut améliorer les performances de votre modèle !
- Une [présentation](#) du transfer learning : *avantages*, *cas d'application*, et un exemple de code avec Tensorflow.
- Comment [mettre en prod](#) un modèle de Deep Learning via une WebApp

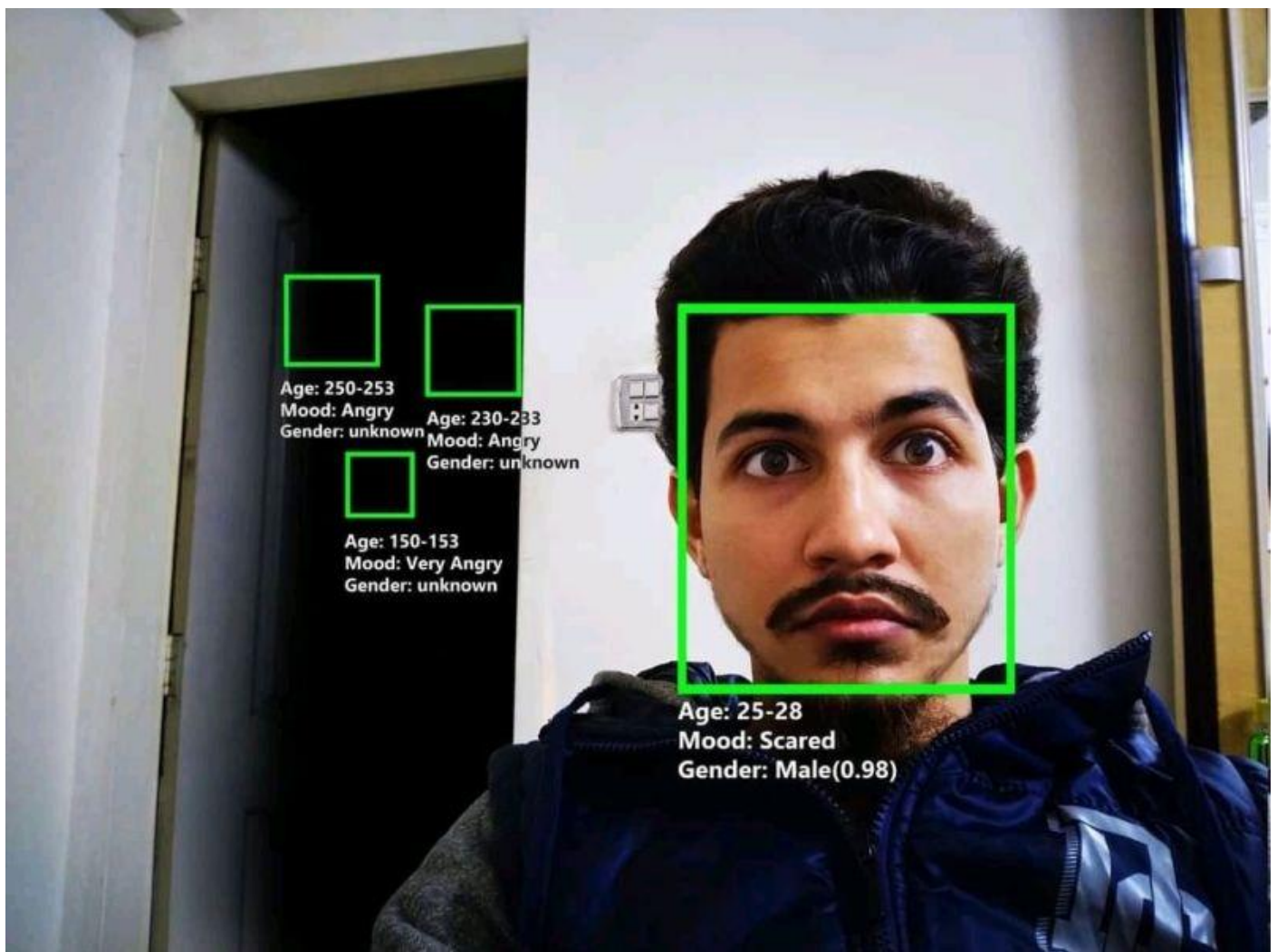
Compétences data à valider

- Se familiariser avec les réseaux de neurones profonds
- Mettre en place un modèle de *Deep Learning* (modèle convolutif)
- Sélectionner un modèle approprié à une problématique métier (Computer Vision)
- Adapter les hyperparamètres du modèle afin de l'améliorer (epochs, batch, activation function, optimizer, ...)
- Évaluer ses performances
- Utiliser le Transfert Learning (GoogleNet, VGG16, mobileNet)
- Intégrer le modèle dans une application/WebApp et la Déployer.

Eventuelles autres sources de données et missions :

<https://cocodataset.org/#home>

When your face recognition algorithm is so advanced that it can even identify spirits in the afterlife.



For those of you who are worried, the story has a happy ending!

