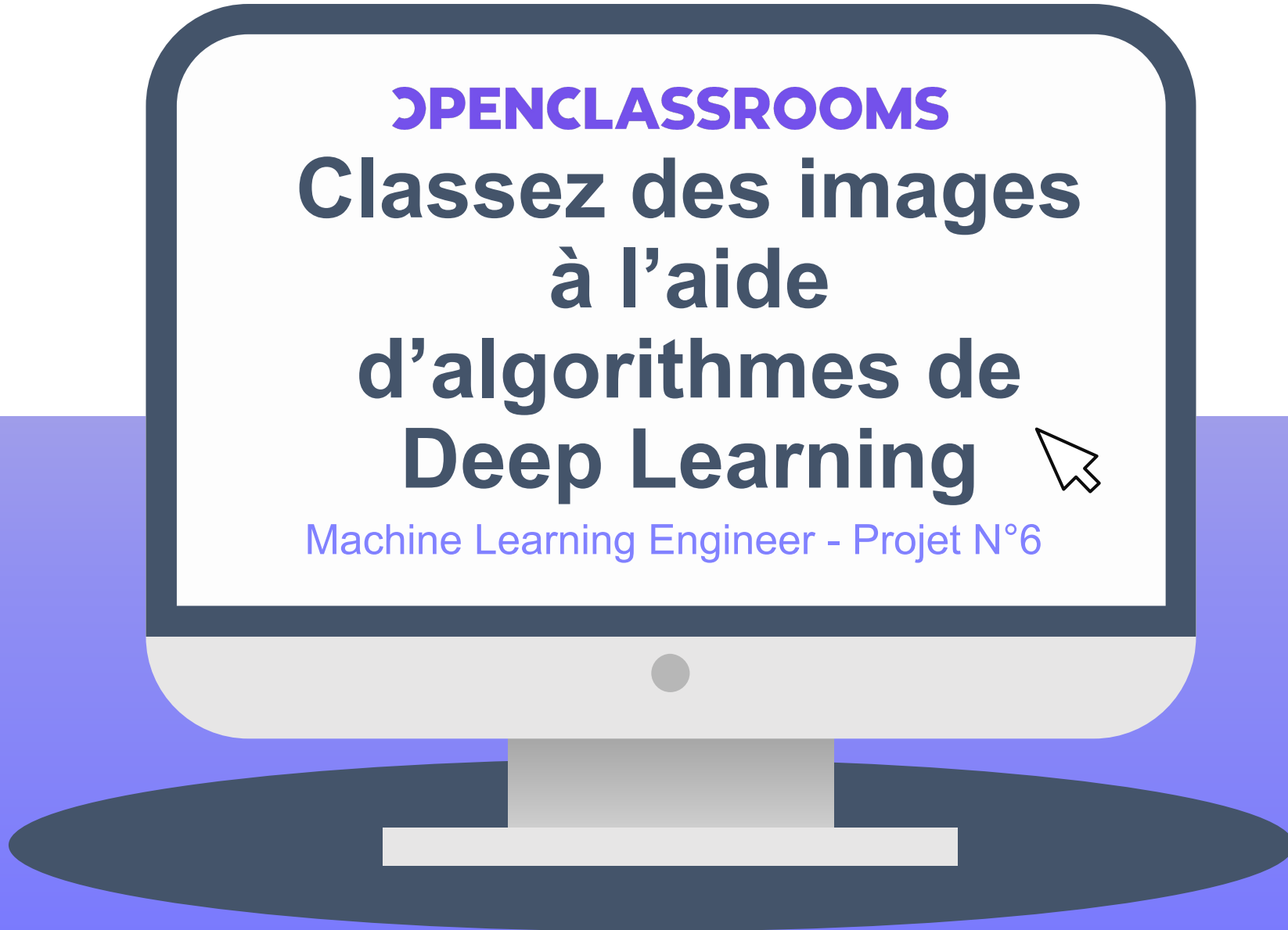


OPENCLASSROOMS







Classez des images à l'aide d'algorithmes de Deep Learning



Machine Learning Engineer - Projet N°6



Sommaire

-  — 01 Contexte de la mission
-  — 02 Collecte, traitement et exploration
-  — 03 Augmentation des données
-  — 04 Modélisation personnelle
-  — 05 Modélisation via Transfer Learning
-  — 06 Démonstration et conclusion

01

Contexte de la mission

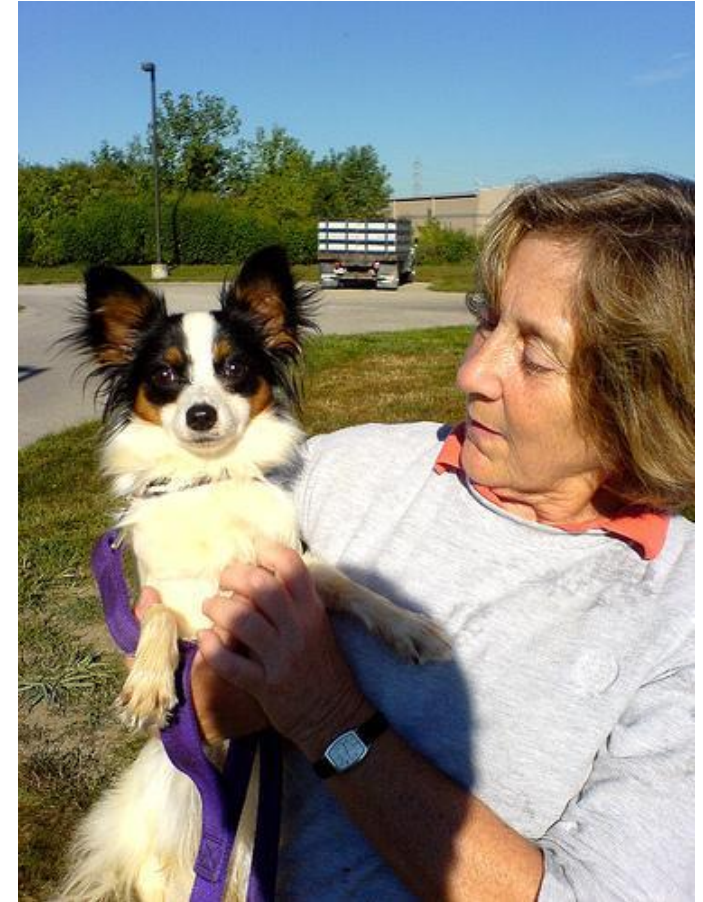


LE CLIENT : LE REFUGE

- Association de protection des animaux, d'où vient mon chat
- Accueil de plus en plus de pensionnaires
- De plus en plus compliqué de maintenir un suivi et une organisation propre des informations

L'OBJECTIF ET LES RESSOURCES

- Mettre en place un algorithme capable d'identifier la race d'un chien à partir d'une image, pour aider le référencement
- Il existe de nombreux datasets publics d'images de chiens
- Essai d'une modélisation "maison", et d'un appui sur la littérature et science des données existantes



02

**Collecte, traitement et
exploration**



COLLECTE DES DONNÉES

- Récupération du très populaire jeu de données **Stanford Dogs Dataset**
- Images extraites de la base de données **ImageNet**
- Plus de 120 races de chien et 20000 images
- Images classées par dossiers, avec certaines métadonnées déjà fournies

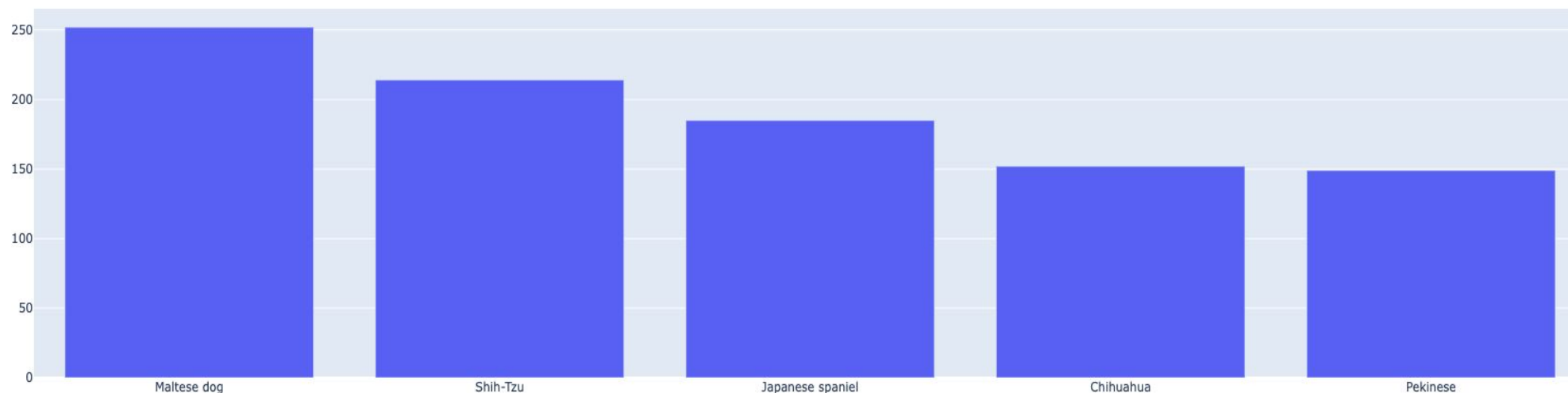


EXPLORATION DES DONNÉES

- Inspection aléatoire d'images de chiens de différentes races pour prendre connaissance du/des formats
- Analyse de la distribution des classes pour vérifier un certain déséquilibre entre les races



Distribution des races



TRAITEMENT DES DONNÉES

Organisation des données

- Conversion des images en *ndarray* pour faciliter les opérations mathématiques
- Création d'un DataFrame pour stocker des métadonnées et faciliter l'analyse et le suivi
 - Chemin du fichier
 - Dimensions
 - Race (nettoyée avec regex)
 - Original / Augmenté
- Échantillonnage avec 5 races pour faciliter la modélisation

Traitement adapté aux données visuelles

- Redimensionnement au format standard (224x224)
- Egalisation des niveaux de couleur
- Whitening
- Normalisation
- Augmentation des données
 - Rotation / Flip
 - Zoom
 - Changement des dimensions

EXEMPLES D'IMAGES AUGMENTÉES

Image originale



03

Modélisation personnelle



PRÉPARATION À LA MODÉLISATION

- **Séparation des données en différents jeux de données**
 - Train pour l'entraînement
 - Utilisation de la validation croisée pour optimiser l'entraînement du modèle
 - Test pour tester le modèle final
- **Encodage des labels de races**
 - Encodage nécessaire pour les CNN
 - Encodage en mode One-Hot



CHOIX DU MODÈLE

Pour classifier des images, le type de modèle le plus adapté est le **Convolutional Neural Network** (CNN), un modèle puissant aux nombreux hyperparamètres

Design de base du modèle

- Objet **Sequential** de la librairie Keras de TensorFlow
- 3 groupes de couches Convolution + Pooling successives
- Méthode d'aggrégation : MaxPooling
- Deux couches denses en sortie avec Relu et SoftMax
- Evaluation sur l'accuracy

Premiers résultats

	Train accuracy	Validation Loss	Validation Accuracy
CNN maison initial	0.95	7.6	0.25

OPTIMISATION DES HYPERPARAMÈTRES

Pour améliorer les résultats de mon modèle personnel, j'ai cherché à optimiser ses hyperparamètres.

Hyperparamètres optimisés

- Test avec plusieurs valeurs d'époques
- Test avec plusieurs valeurs de nombre de couches de convolution & pooling
- RandomSearch sur :
 - Le nombre de filtres des couches de convolution
 - Le nombre de neurones des couches denses
 - Le taux d'apprentissage

Premiers résultats

	Train accuracy	Validation Loss	Validation Accuracy
CNN maison initial	0.95	7.6	0.25
CNN maison opti	0.95	7.2	0.26

Il y'a clairement **overfitting** quand j'entraîne mon CNN à partir des données de Stanford uniquement, et le modèle n'arrive pas à généraliser.

04

Transfer Learning

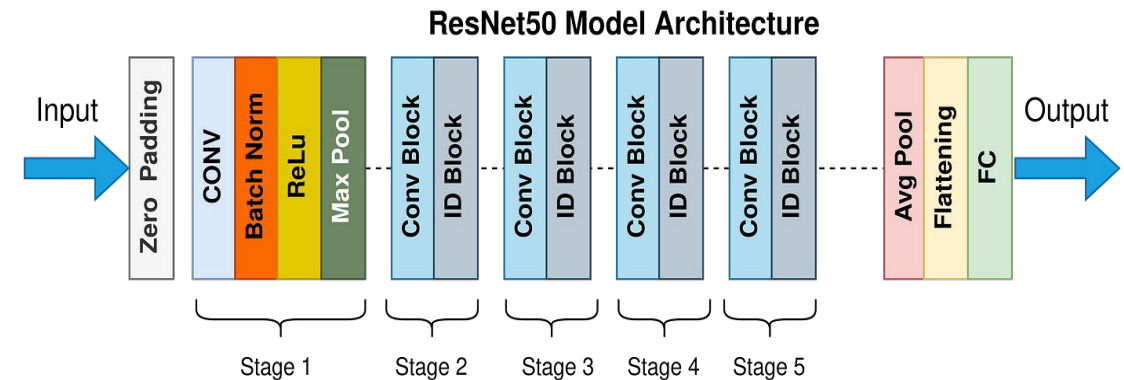


CHOIX DES MODÈLES DE TRANSFER LEARNING

Les résultats obtenus sur les modèles non pré-entraînés ne sont pas satisfaisants. Je teste donc l'utilisation de modèles pré-entraînés.

Choix des modèles

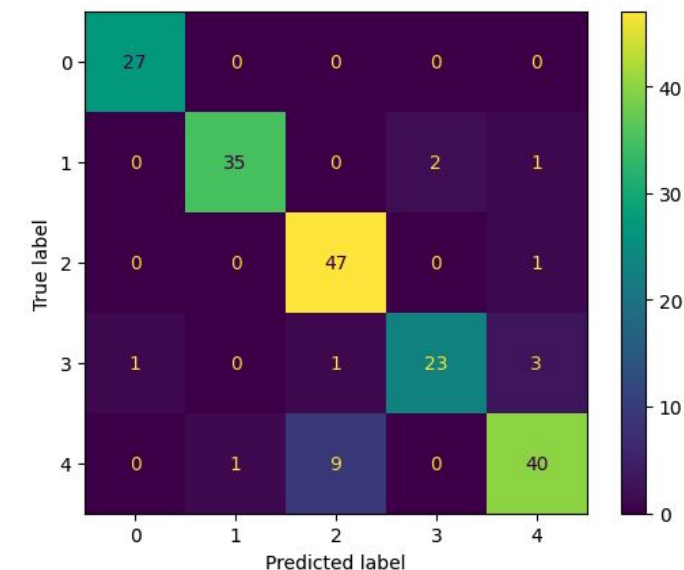
- **VGG16**
 - Modèle simple basée sur l'architecture d'un réseau à convolution classique, mais volumineux
 - Entraîné sur les images d'ImageNet
- **Resnet50**
 - Modèle basé sur l'architecture des réseaux résiduels
 - Permet une meilleure conservation de l'information entre les couches du CNN
 - Entraîné sur les images d'ImageNet



COMPARAISON DES RÉSULTATS

	Train accuracy	Validation Loss	Validation Accuracy
CNN maison initial	95%	7.6	25%
CNN maison optimisé	95%	7.2	26%
VGG16	78%	1.04	68%
ResNet50	95%	0.4	90%

L'approche par Transfer Learning est bien plus concluante. ResNet50 est précis à **90%** dans ses prédictions.



*Matrice de confusion
avec ResNet50*

05

Démo

