OPENCLASSROOMS

PROJET N°6

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

Guillaume Lemêle - *Machine Learning Engineer* -10/2022 – 06/2023

SOMMAIRE

CLASSEZ DES IMAGES À L'AIDE D'ALGORITHMES DE DEEP LEARNING

Étape n° 1

- Présentation de la problématique et du prétraitement effectué, cleaning, feature engineering et EDA

Étape n° 2

- Présentation des différentes pistes de modélisation effectuées

Étape n° 3

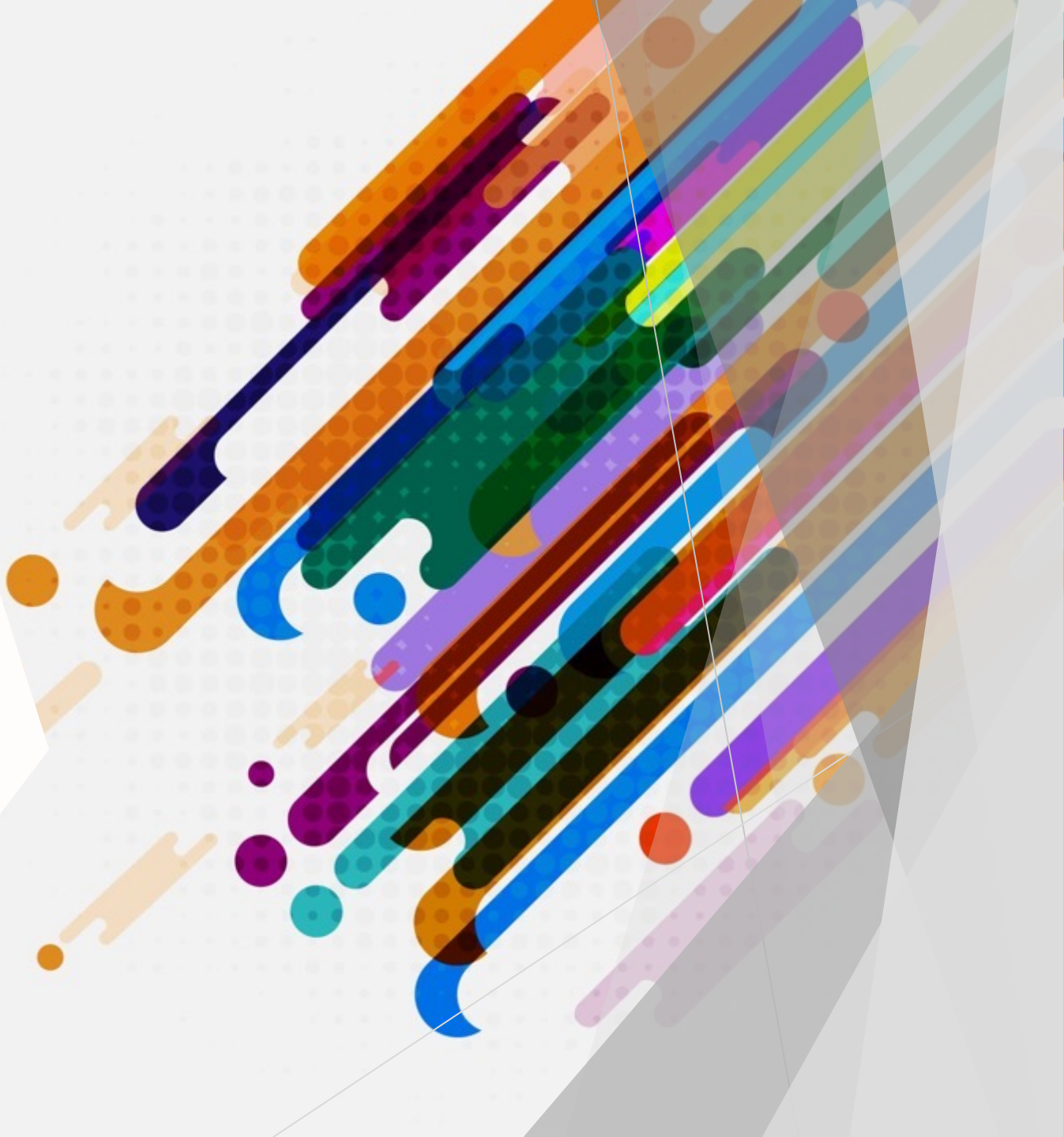
- Choix du modèle final sélectionné (pour chaque approche) et améliorations effectuées

Étape n° 4

- Démonstration du programme

Présentation de la problématique et du prétraitement effectué, cleaning, feature engineering et exploration

ÉTAPE N° 1



Présentation de la problématique

- Problématique :

- Vous êtes bénévole pour l'association de protection des animaux de votre quartier. Leur base de données 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.

- Objectif :

- Obtenir un algorithme capable de classer les images en fonction de la race du chien présent sur l'image.

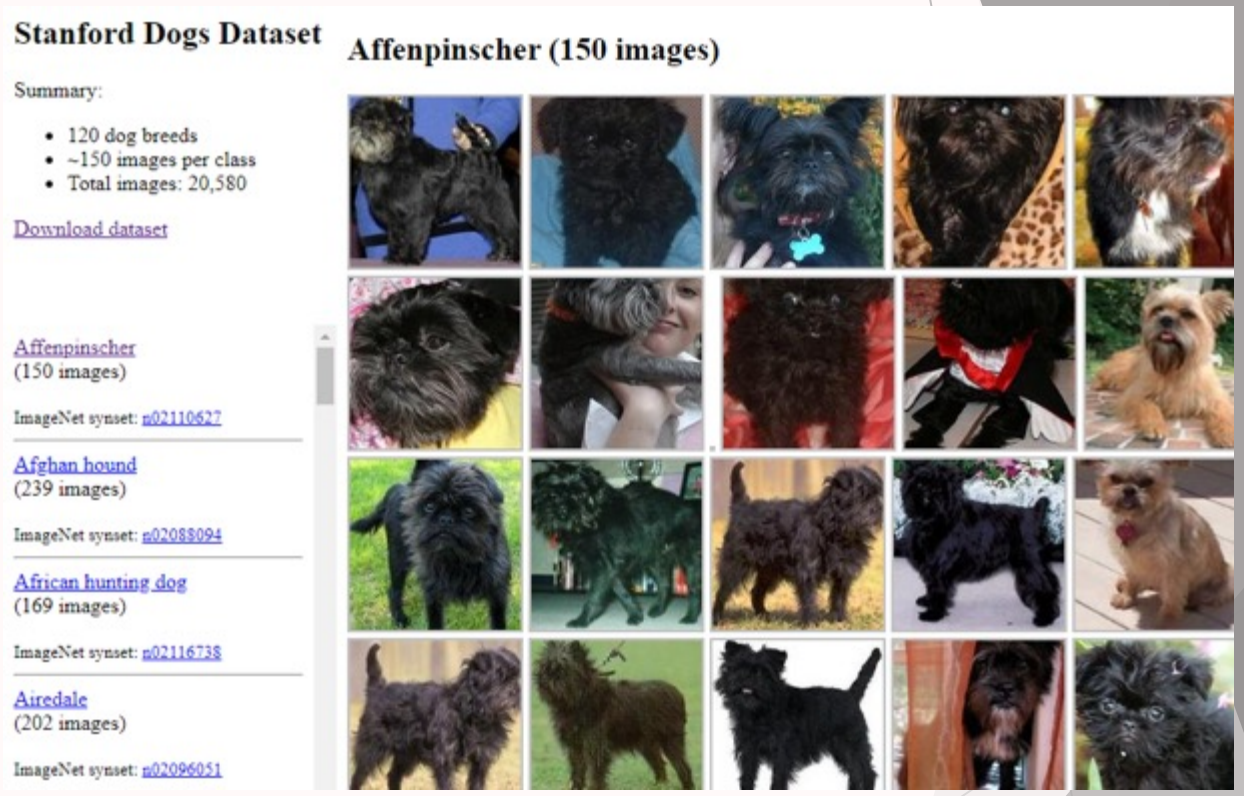
- Mission :

- **Pré-processer des images avec des techniques spécifiques**
- **Réaliser un algorithme de détection de la race du chien**

Récupération de la base de données

- **Stanford Dogs Dataset**

- **Number of categories:** 120
- **Number of images:** 20,580



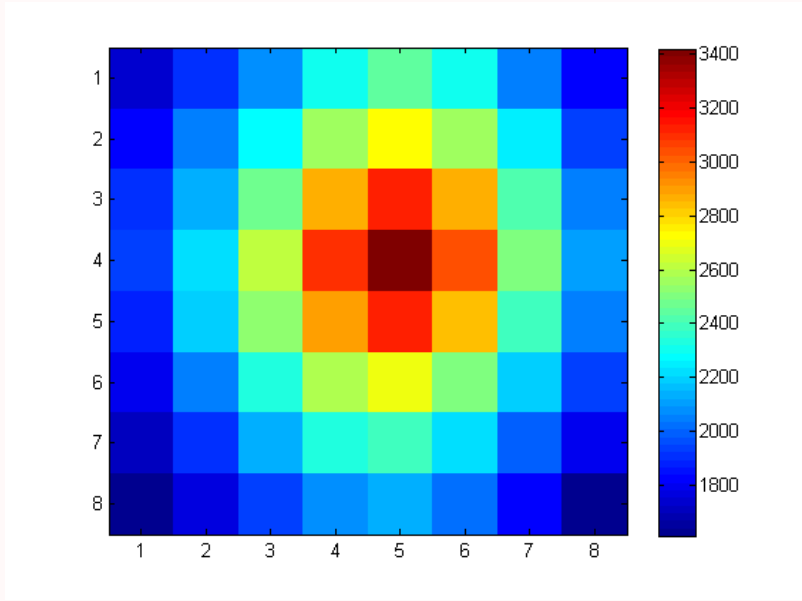
Prétraitement effectué

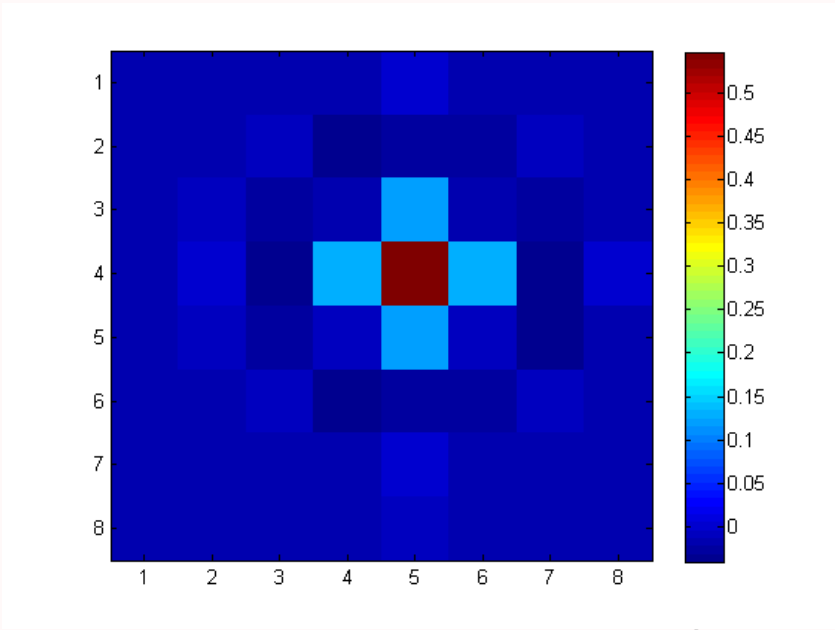
Filtrage des 15 races
les plus communément
présentes dans le dataset.

Breed	Number of Images
n02085936-Maltese_dog	252
n02088094-Afghan_hound	239
n02092002-Scottish_deerhound	232
n02112018-Pomeranian	219
2107683-Bernese_mountain_dog	218
n02111889-Samoyed	218
n02090721-Irish_wolfhound	218
n02086240-Shih-Tzu	214
n02111500-Great_Pyrenees	213
n02111129-Leonberg	210
n02110806-basenji	209
n02097474-Tibetan_terrier	206
n02095889-Sealyham_terrier	202
n02108000-EntleBucher	202
n02096051-Airedale	202

Preprocessing : Whitening

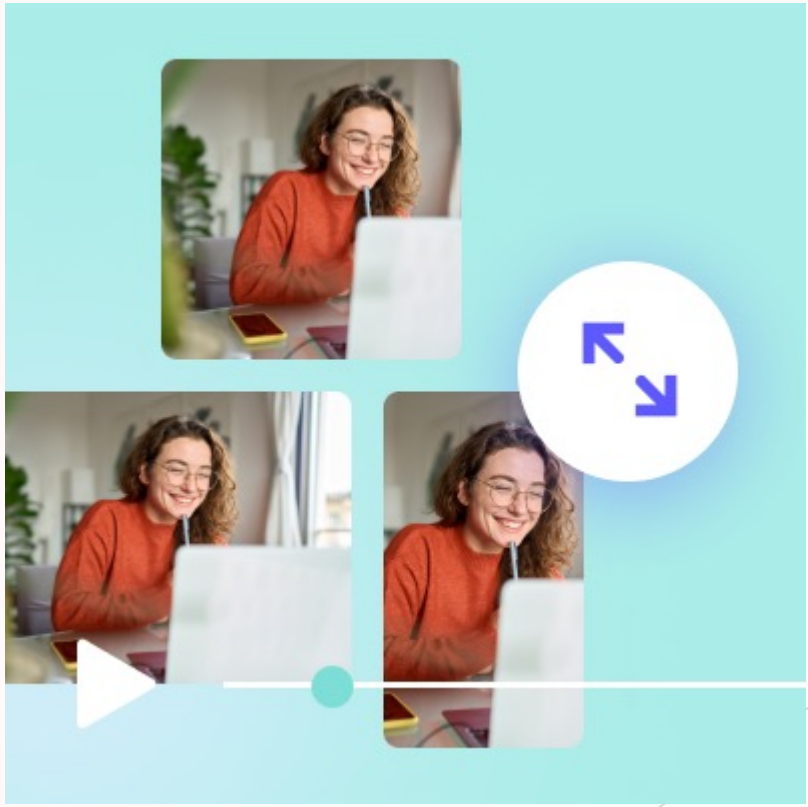
- Supprime les corrélations entre les différentes caractéristiques des données et égalise leur importance
- Plus vulgairement ; sert à rendre les données plus "nettes" et plus faciles à analyser pour les algorithmes. Va « comme » éclaircir les couleurs et normaliser les tailles.





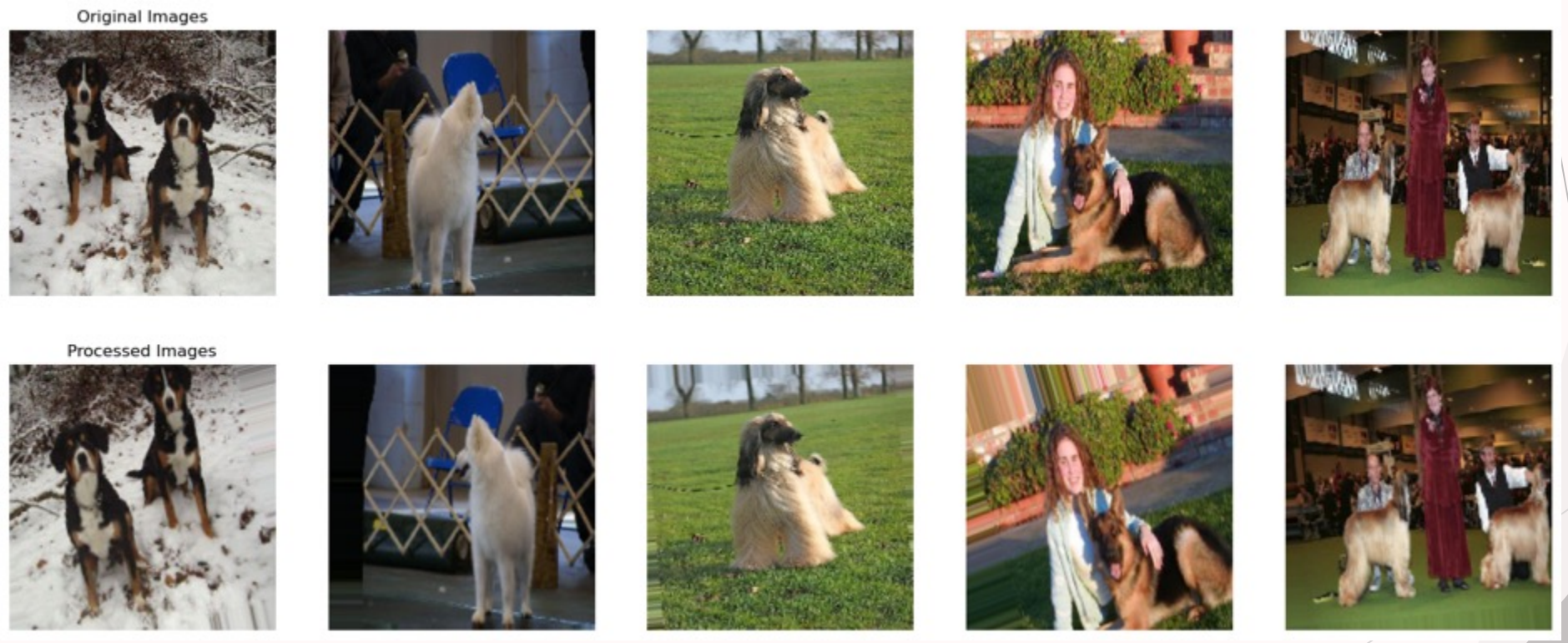
Preprocessing : Resizing

- Redimensionner toutes les images à une taille cohérente (dans notre cas, nous choisissons 180x180 pixels).

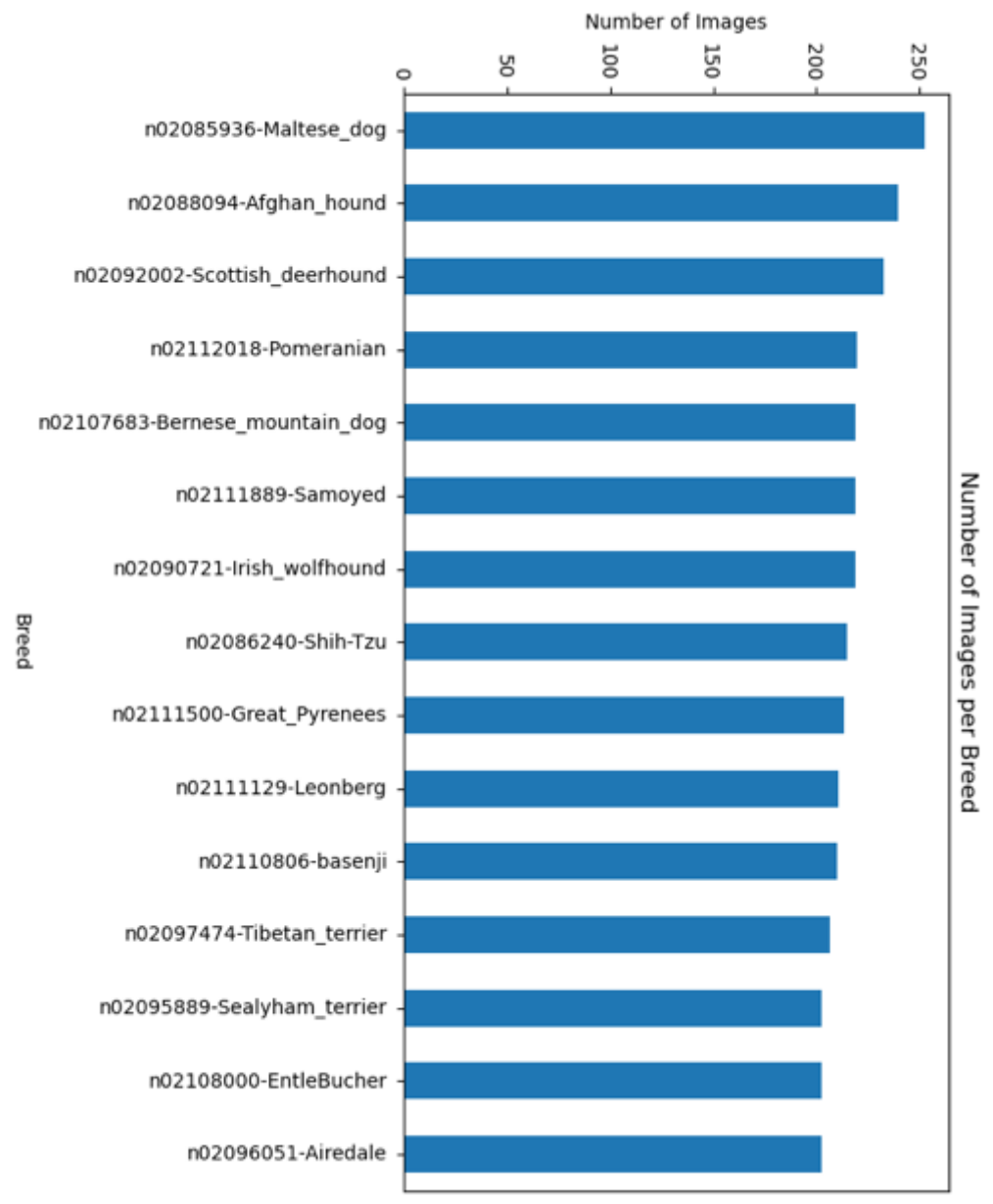


Preprocessing : Data Augmentation

- Mise en miroir : horizontal_flip
- Rotation, zoom et décalage



EDA


Number of Images
0
50
100
150
200
250
n02085936-Maltese_dog
n02088094-Afghan_hound
n02092002-Scottish_deerhound
n02112018-Pomeranian
n02107683-Bernese_mountain_dog
n02111889-Samoyed
n02090721-Irish_wolfhound
n02086240-Shih-Tzu
n02111500-Great_Pyrenees
n02111129-Leonberg
n02110806-basenji
n02097474-Tibetan_terrier
n02095889-Sealyham_terrier
n02108000-EntleBucher
n02096051-Airedale
Breed
Number of Images per Breed

Présentation des différentes pistes de modélisation effectuées

ÉTAPE N° 2

Présentation des différentes pistes de modélisation effectuées

Nous appliquons 2 approches à notre projet :

1. Une première **en réalisant notre propre réseau CNN**, en nous inspirant de réseaux CNN existants. Nous prenons soin d'optimiser certains hyperparamètres (des layers du modèle, de la compilation du modèle et de l'exécution du modèle)

2. Une deuxième **en utilisant le transfert learning**, c'est-à-dire en utilisant un réseau déjà entraîné, et en le modifiant pour répondre à notre problématique.

Process

1. Séparation du jeu de données
   - Création d'un ensemble d'entraînement (train set) et d'un ensemble de test (test set de 20%)

2. Division stratifiée des ensembles d'entraînement et de test.
   - Garder la même proportion des données selon la race de chien.

3. Initialiser ImageDataGenerator avec Keras.

4. Construction du CNN (nombres de filtres à convolutions, taille du kernel, etc). Compilation et exécution avec Keras.

5. Sélection et recherche des meilleurs hyparamètres et itérations successives.
   - Utilisation RandomSearchCV pour l'optimisation des hyperparamètres
   - Différents tuners; espaces de recherches. Early stopping et régularisation.

6. Transfert learning : choix du modèle Xception model car il demande moins de ressources

Process

7. Choix du meilleur modèle
- a) Accuracy
- b) Precision, recall et f1 score

- Présentation pour chaque modèle d'une matrice de classification
- Analyse de la précision et du rappel pour évaluer la qualité des prédictions.

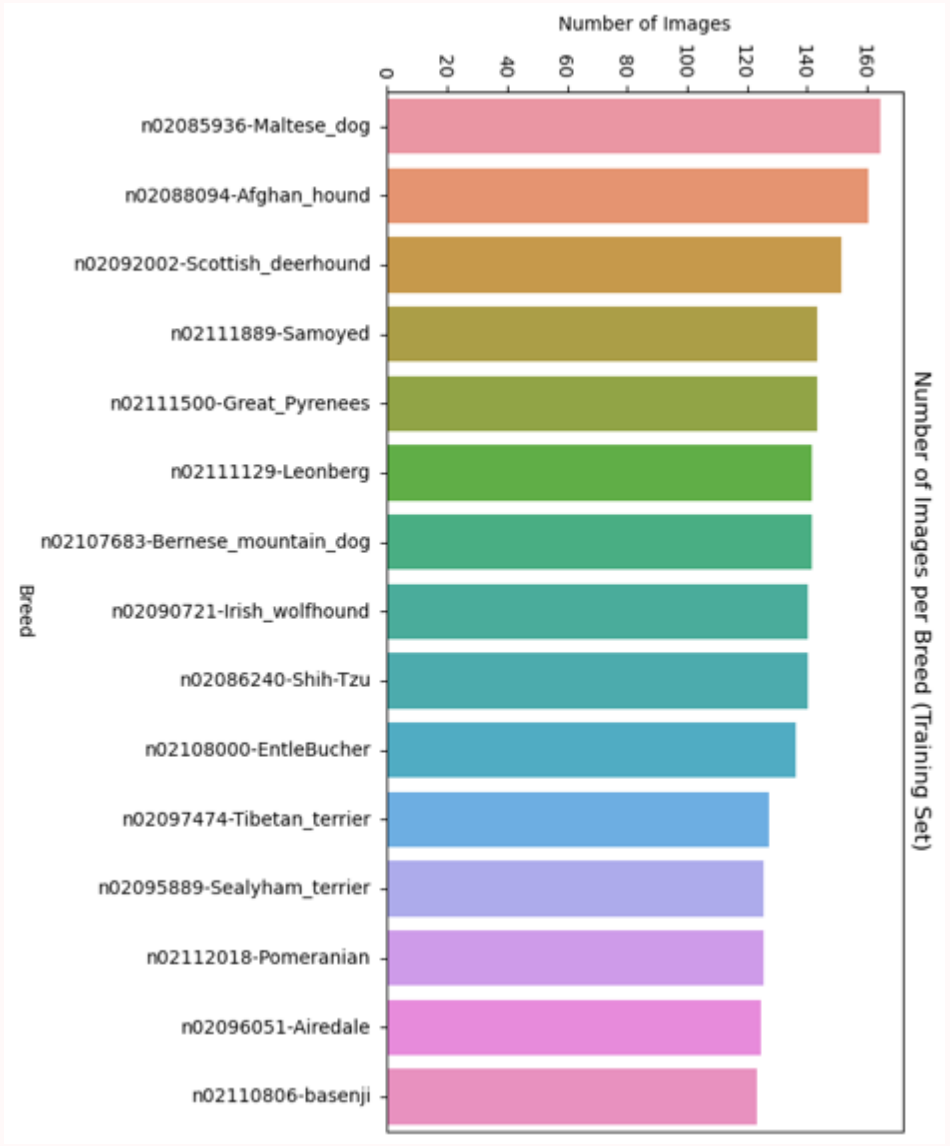
Le rappel mesure la proportion de vrais positifs qui sont correctement identifiées par le modèle

La précision mesure la proportion de vrais positifs qui sont réellement positifs

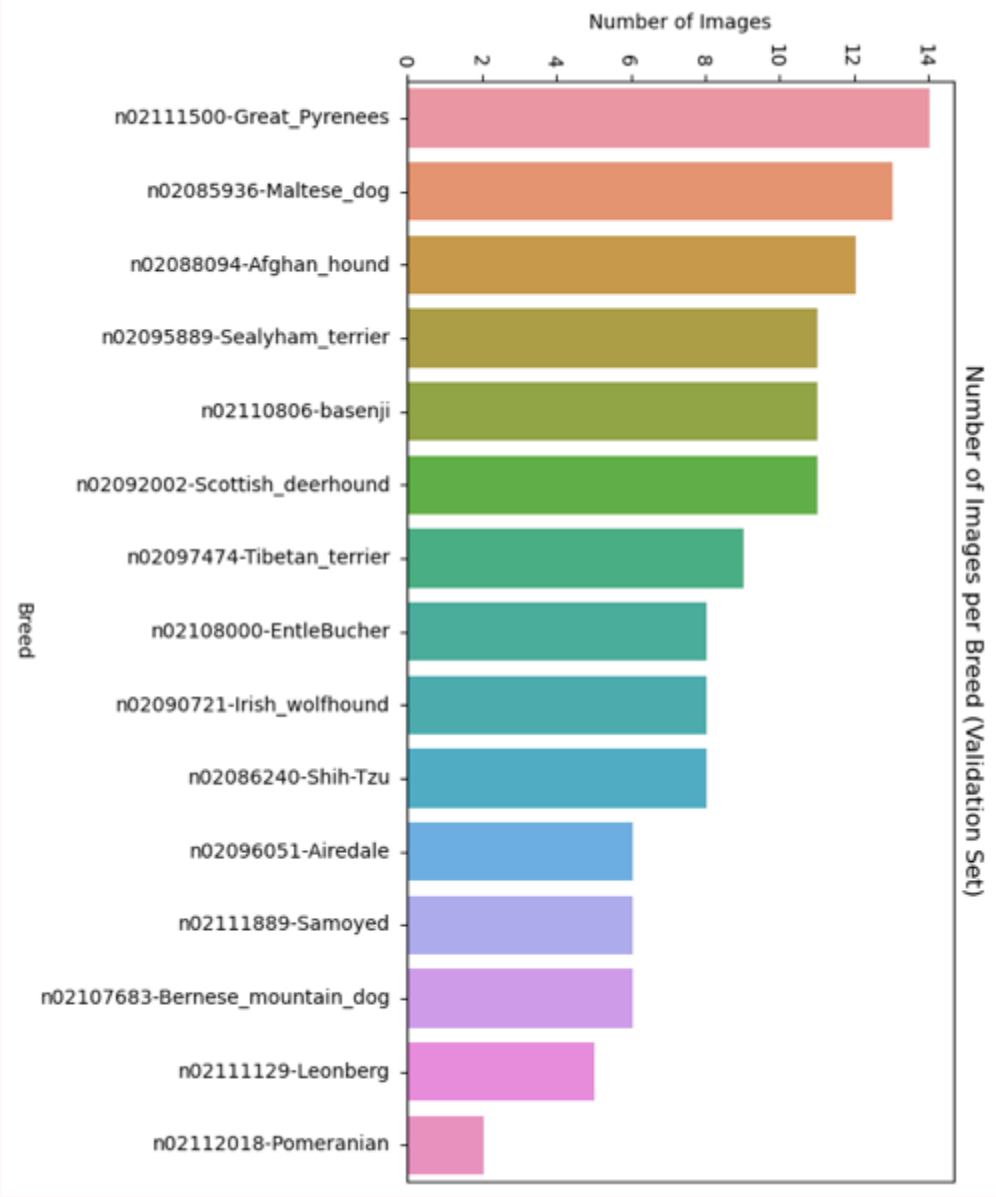
- F1 score : moyenne harmonique de la précision et du rappel

- Visualisation d'une matrice de confusion

Process

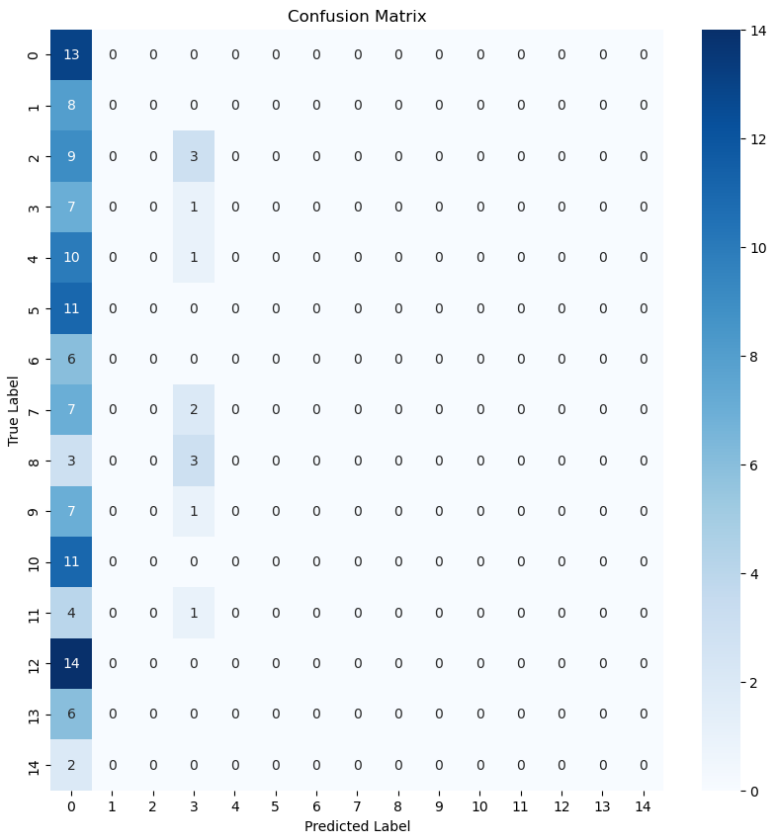

Number of Images
0
20
40
60
80
100
120
140
160
n02085936-Maltese_dog
n02088094-Afghan_hound
n02092002-Scottish_deerhound
n02111889-Samoyed
n02111500-Great_Pyrenees
n02111129-Leonberg
n02107683-Bernese_mountain_dog
n02090721-Irish_wolfhound
n02086240-Shih-Tzu
n02108000-EntleBucher
n02097474-Tibetan_terrier
n02095889-Sealyham_terrier
n02112018-Pomeranian
n02096051-Airedale
n02110806-basenji
Breed
Number of Images per Breed (Training Set)

Process


Number of Images
0
2
4
6
8
10
12
14
n02111500-Great_Pyrenees
n02085936-Maltese_dog
n02088094-Afghan_hound
n02095889-Sealyham_terrier
n02110806-basenji
n02092002-Scottish_deerhound
n02097474-Tibetan_terrier
n02108000-EntleBucher
n02090721-Irish_wolfhound
n02086240-Shih-Tzu
n02096051-Airedale
n02111889-Samoyed
n02107683-Bernese_mountain_dog
n02111129-Leonberg
n02112018-Pomeranian
Breed
Number of Images per Breed (Validation Set)

Modèle from scratch : premières itérations

- Surentrainement malgré la recherche in-fine de différents hyperparamètres.



Found 130 images belonging to 15 classes.

	precision	recall	f1-score	support
0	0.11	1.00	0.20	13
1	0.00	0.00	0.00	8
2	0.00	0.00	0.00	12
3	0.08	0.12	0.10	8
4	0.00	0.00	0.00	11
5	0.00	0.00	0.00	11
6	0.00	0.00	0.00	6
7	0.00	0.00	0.00	9
8	0.00	0.00	0.00	6
9	0.00	0.00	0.00	8
10	0.00	0.00	0.00	11
11	0.00	0.00	0.00	5
12	0.00	0.00	0.00	14
13	0.00	0.00	0.00	6
14	0.00	0.00	0.00	2
accuracy			0.11	130
macro avg	0.01	0.07	0.02	130
weighted avg	0.02	0.11	0.03	130

Modèle from scratch : Random Search

- Meilleurs hyperparamètres :

The optimal learning rate for the optimizer is 0.0001.

The optimal number of filters in the first conv layer is 96

The optimal size of the kernel in the first conv layer is 5

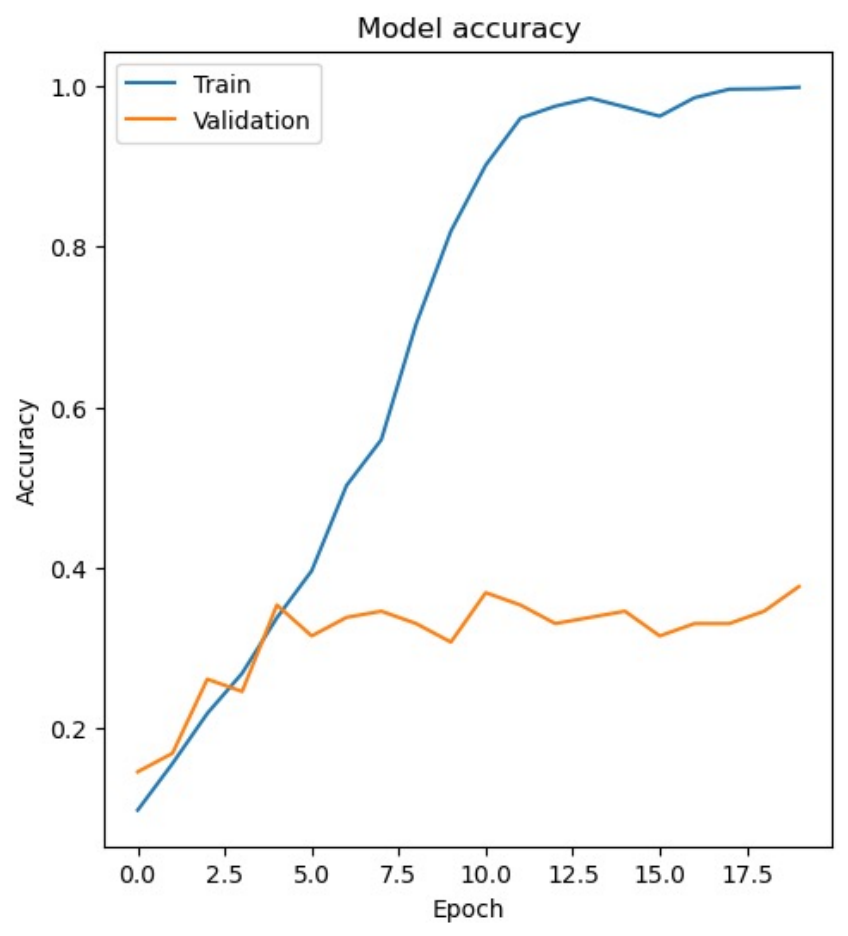
The optimal number of filters in the second conv layer is 32

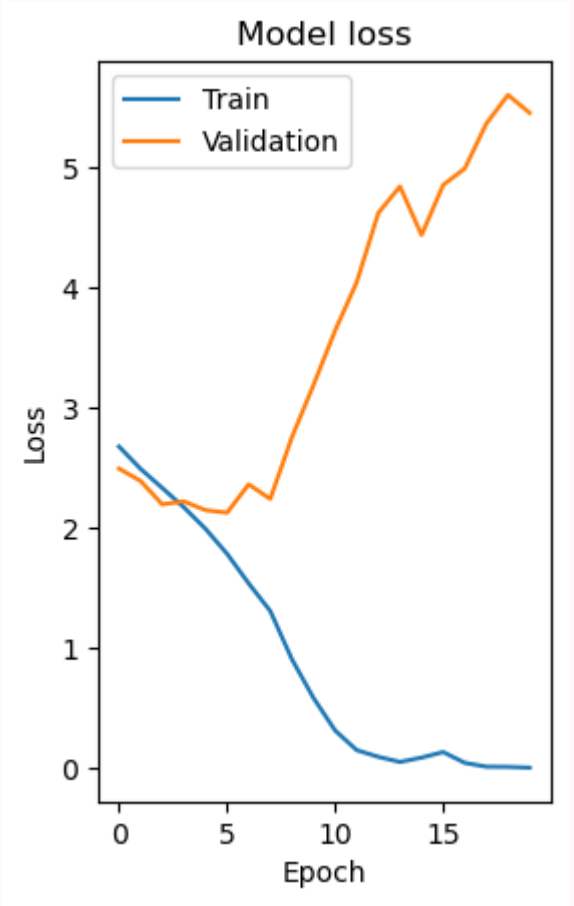
The optimal size of the kernel in the second conv layer is 5

Found 130 images belonging to 15 classes.

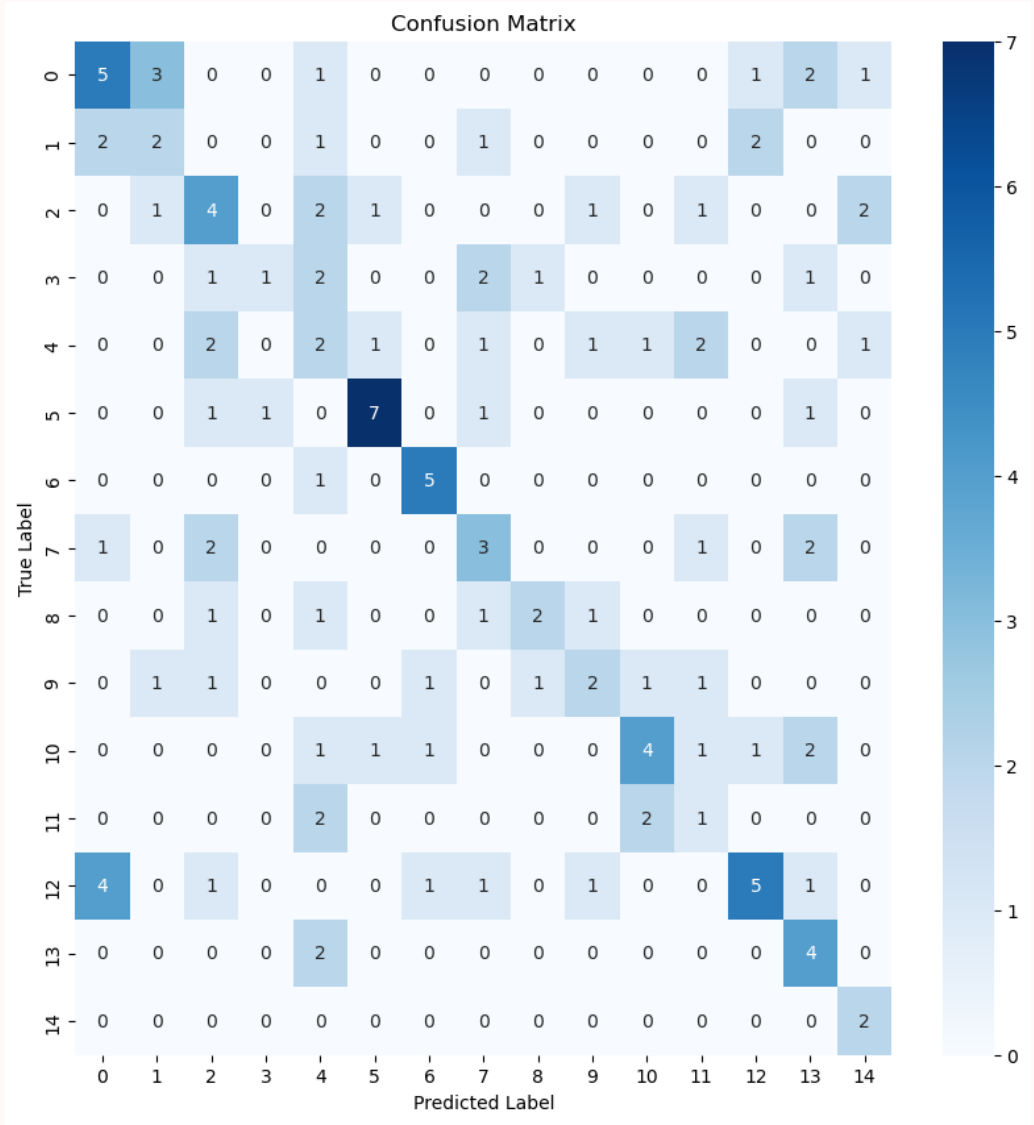
	precision	recall	f1-score	support
n02085936-Maltese_dog	0.42	0.38	0.40	13
n02086240-Shih-Tzu	0.29	0.25	0.27	8
n02088094-Afghan_hound	0.31	0.33	0.32	12
n02090721-Irish_wolfhound	0.50	0.12	0.20	8
n02092002-Scottish_deerhound	0.13	0.18	0.15	11
n02095889-Sealyham_terrier	0.70	0.64	0.67	11
n02096051-Airedale	0.62	0.83	0.71	6
n02097474-Tibetan_terrier	0.30	0.33	0.32	9
n02107683-Bernese_mountain_dog	0.50	0.33	0.40	6
n02108000-EntleBucher	0.33	0.25	0.29	8
n02110806-basenji	0.50	0.36	0.42	11
n02111129-Leonberg	0.14	0.20	0.17	5
n02111500-Great_Pyrenees	0.56	0.36	0.43	14
n02111889-Samoyed	0.31	0.67	0.42	6
n02112018-Pomeranian	0.33	1.00	0.50	2
accuracy			0.38	130
macro avg	0.40	0.42	0.38	130
weighted avg	0.41	0.38	0.37	130

Modèle from scratch : Random Search





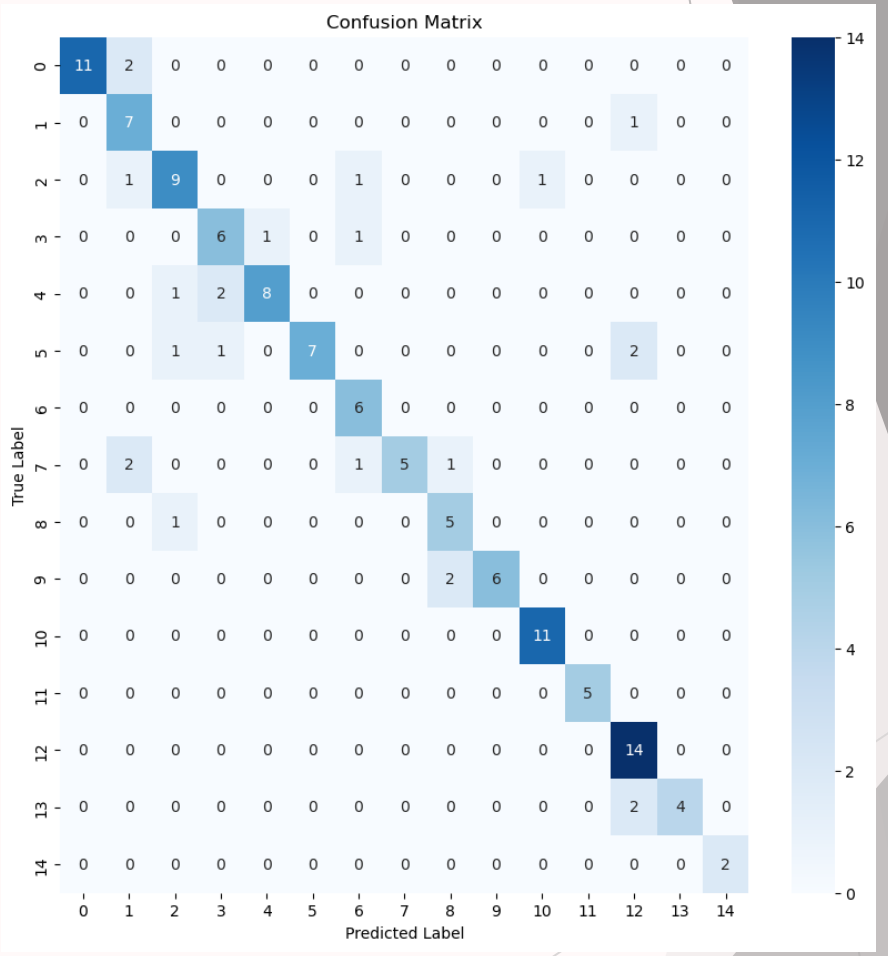
Modèle from scratch : Random Search



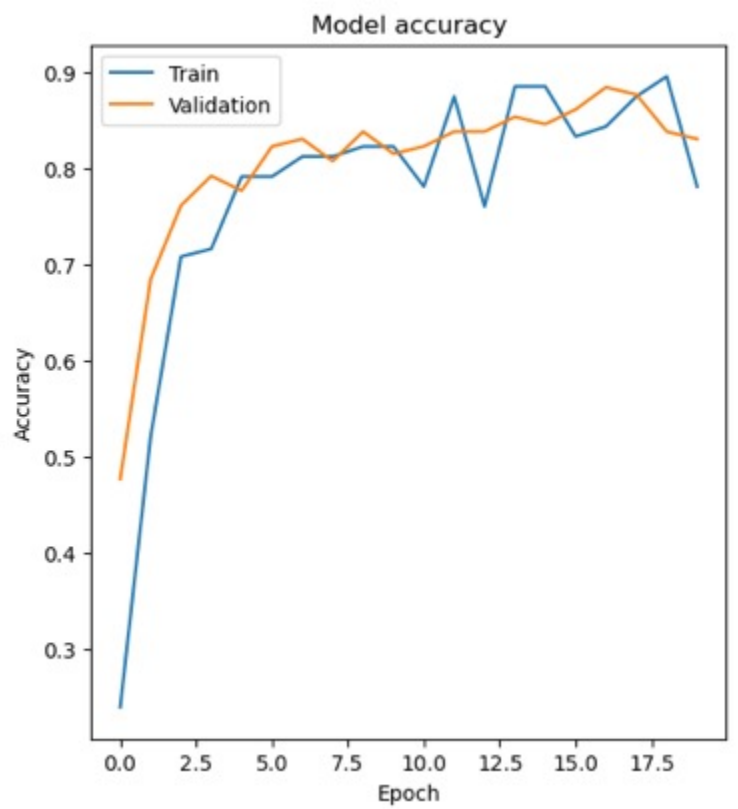
Modèle transfert learning : Xception

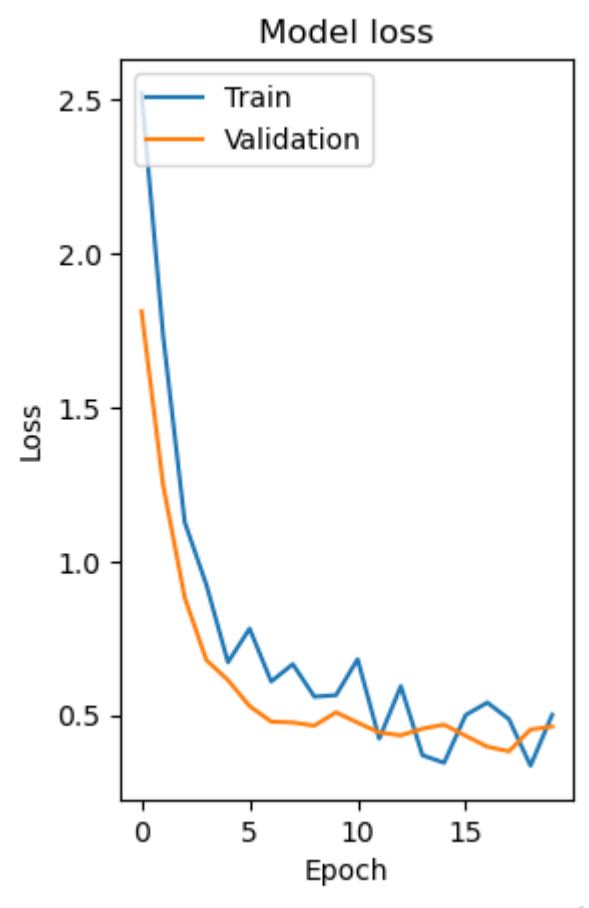
Found 130 images belonging to 15 classes.

	precision	recall	f1-score	support
0	1.00	0.85	0.92	13
1	0.58	0.88	0.70	8
2	0.75	0.75	0.75	12
3	0.67	0.75	0.71	8
4	0.89	0.73	0.80	11
5	1.00	0.64	0.78	11
6	0.67	1.00	0.80	6
7	1.00	0.56	0.71	9
8	0.62	0.83	0.71	6
9	1.00	0.75	0.86	8
10	0.92	1.00	0.96	11
11	1.00	1.00	1.00	5
12	0.74	1.00	0.85	14
13	1.00	0.67	0.80	6
14	1.00	1.00	1.00	2
accuracy			0.82	130
macro avg	0.86	0.83	0.82	130
weighted avg	0.85	0.82	0.82	130



Modèle transfert learning : Xception





Choix du modèle final sélectionné (pour chaque approche) et améliorations effectuées

ÉTAPE N° 3

Modèle final sélectionné

Nous appliquons 2 approches à notre projet :

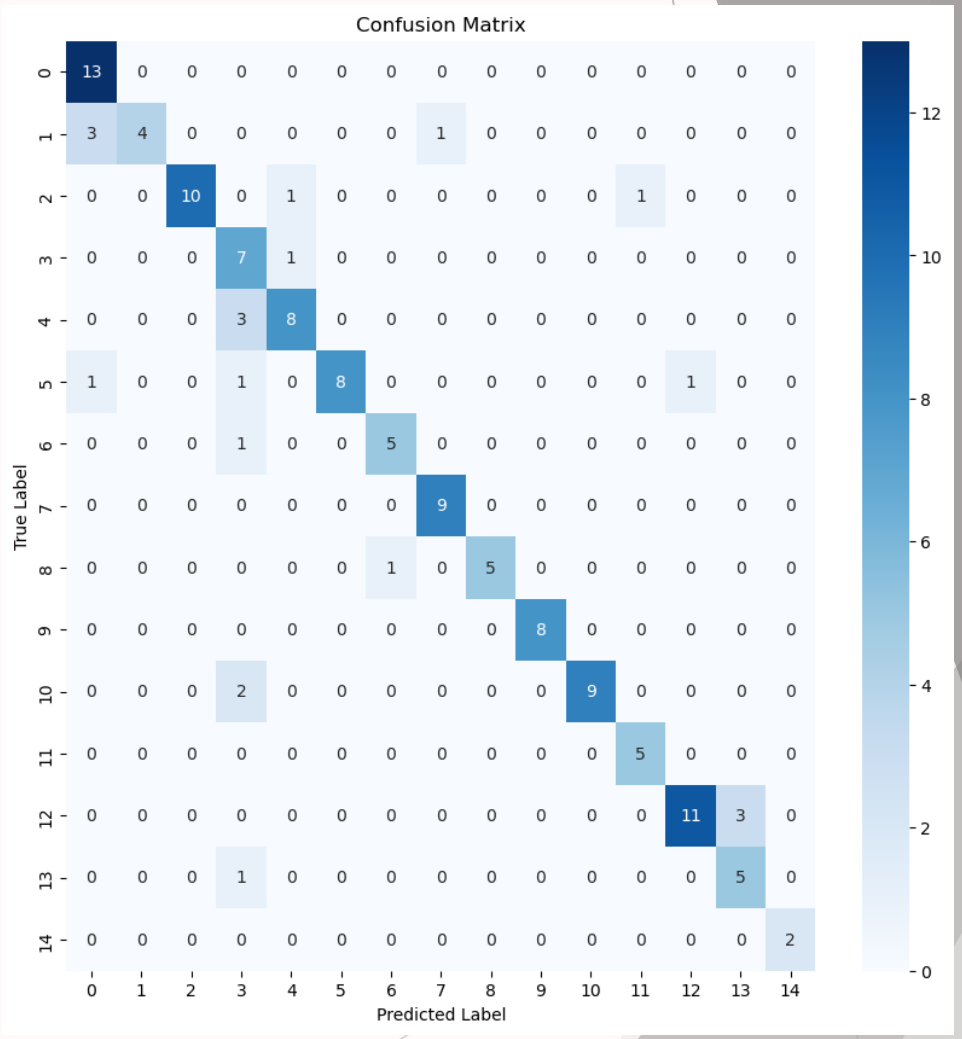
1. Pour la première approche, le modèle final sélectionné est celui issue de notre Random Search.

2. Pour la deuxième approche, le modèle final sélectionné est celui du modèle Xception

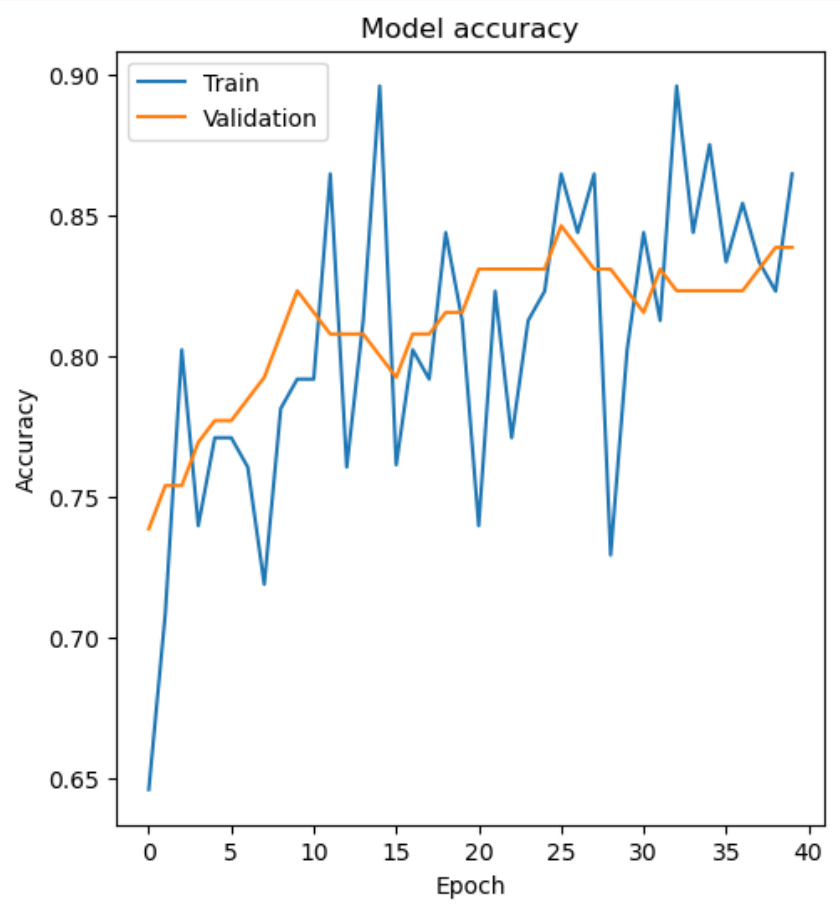
Performances et améliorations effectuées
Modèle transfert learning : learning rate optimization

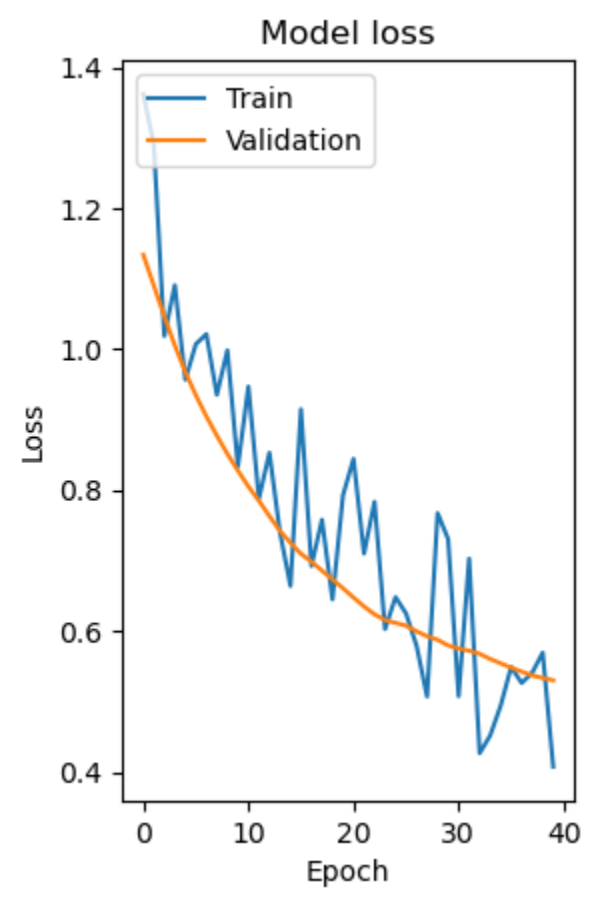
Found 130 images belonging to 15 classes.

	precision	recall	f1-score	support
0	0.76	1.00	0.87	13
1	1.00	0.50	0.67	8
2	1.00	0.83	0.91	12
3	0.47	0.88	0.61	8
4	0.80	0.73	0.76	11
5	1.00	0.73	0.84	11
6	0.83	0.83	0.83	6
7	0.90	1.00	0.95	9
8	1.00	0.83	0.91	6
9	1.00	1.00	1.00	8
10	1.00	0.82	0.90	11
11	0.83	1.00	0.91	5
12	0.92	0.79	0.85	14
13	0.62	0.83	0.71	6
14	1.00	1.00	1.00	2
accuracy			0.84	130
macro avg	0.88	0.85	0.85	130
weighted avg	0.88	0.84	0.84	130



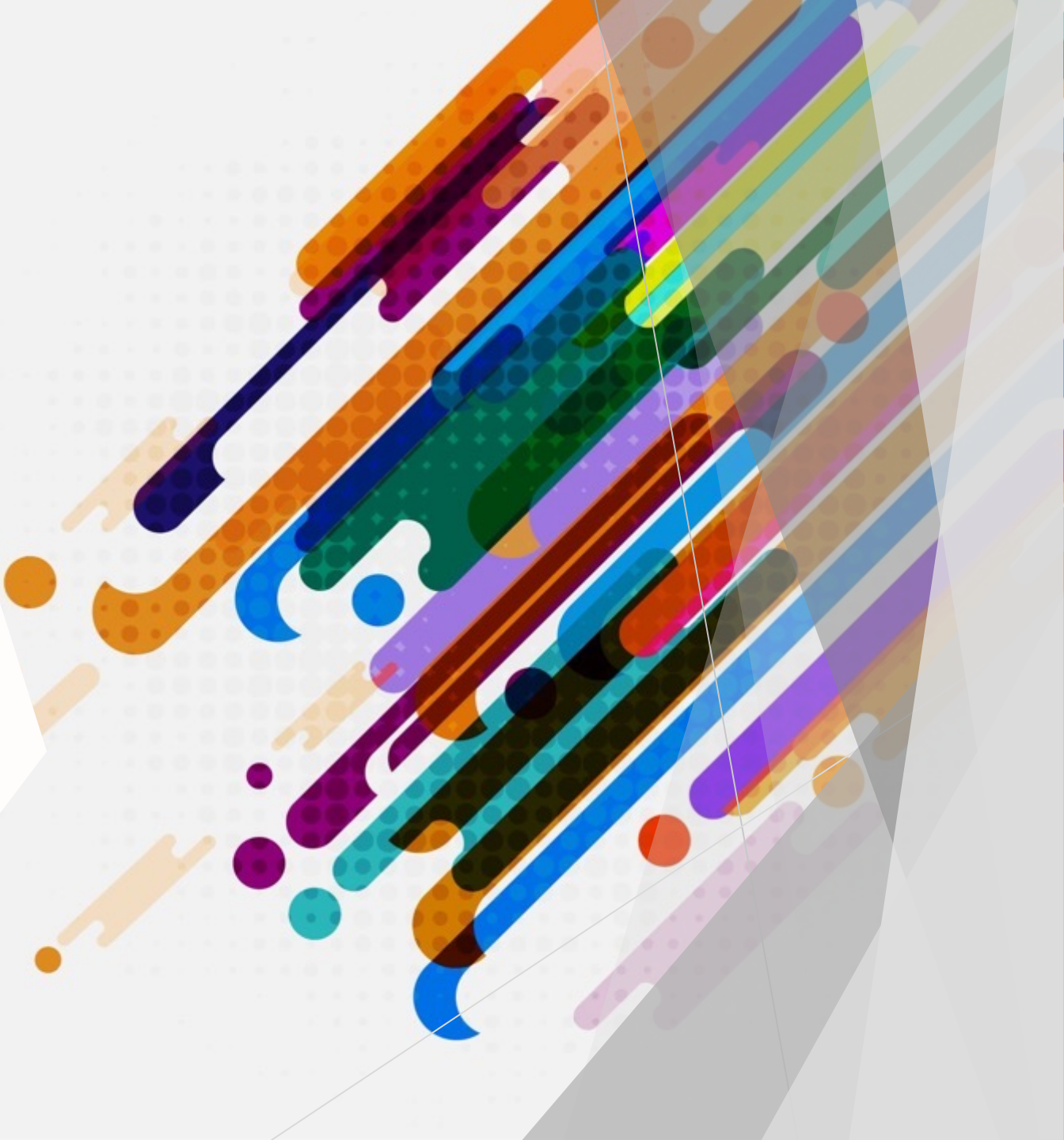
Performances et améliorations effectuées
Modèle transfert learning : Xception





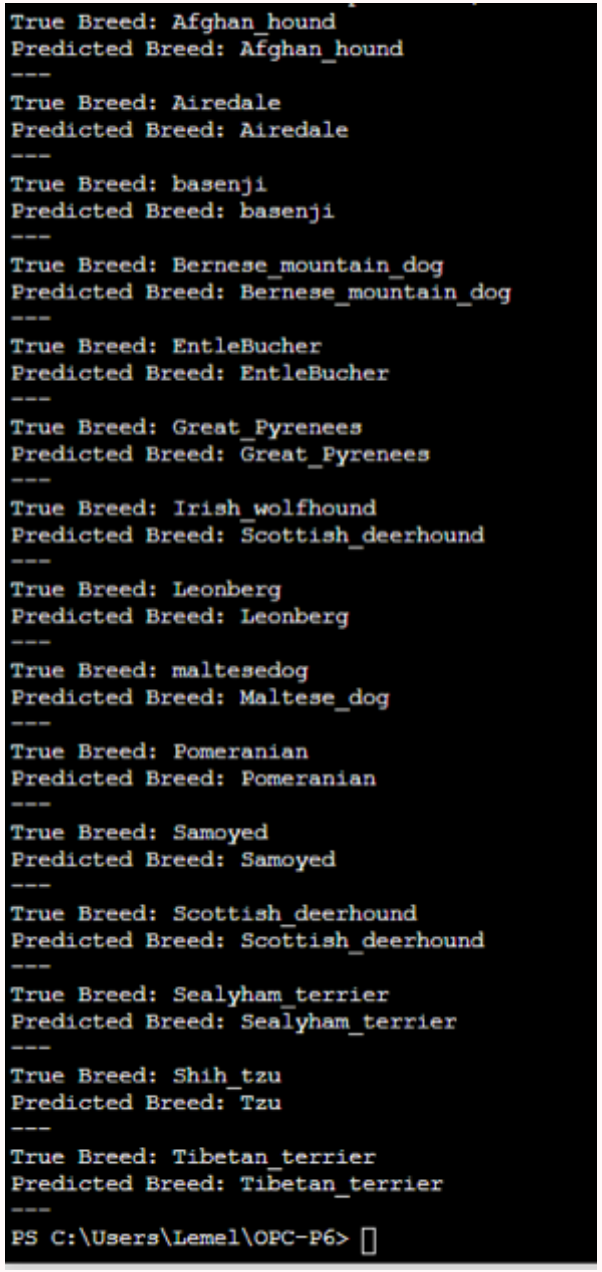
Démonstration du programme

ÉTAPE N° 4



Démonstration du programme : exemple de prédiction

Un **programme** Python qui prend une image (array) en entrée de chaque race de chien où notre meilleur modèle est entraîné et retourne la race la plus probable du chien présent sur l'image.



ALLER PLUS LOIN

- **Point n° 1**

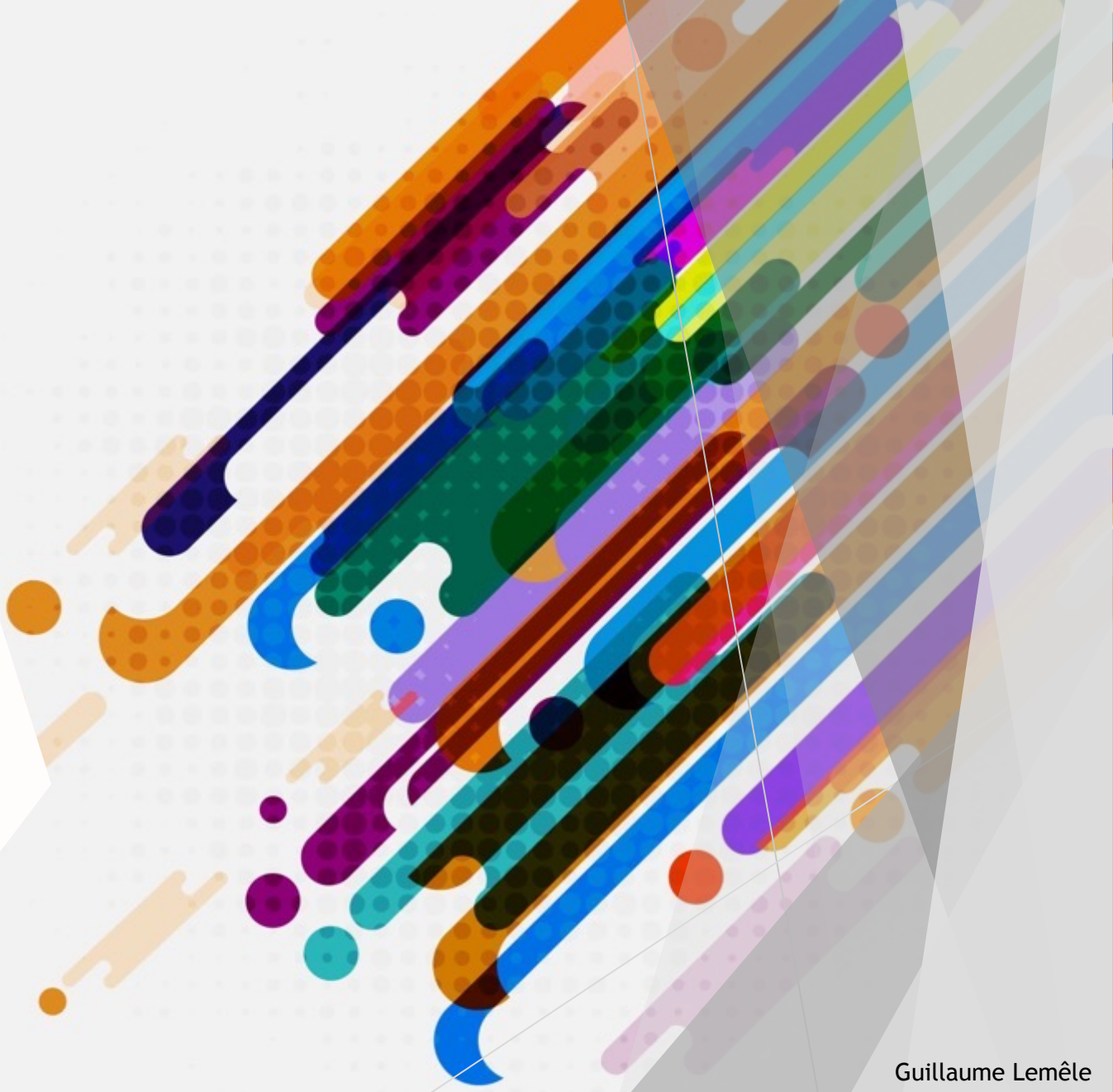
Amélioration du pré-processing

- **Point n° 2**

Amélioration des performances des modèles

DISCUSSION

MERCI POUR VOTRE ÉCOUTE !



Guillaume Lemêle

Parcours : Machine Learning Engineer

10/2022 – 06/2023